



HAL
open science

Innovative blockchain-based farming marketplace and smart contract performance evaluation

Guilain Leduc, Sylvain Kubler, Jean-Philippe Georges

► **To cite this version:**

Guilain Leduc, Sylvain Kubler, Jean-Philippe Georges. Innovative blockchain-based farming marketplace and smart contract performance evaluation. *Journal of Cleaner Production*, 2021, 306, pp.127055. 10.1016/j.jclepro.2021.127055 . hal-03240813

HAL Id: hal-03240813

<https://hal.science/hal-03240813v1>

Submitted on 28 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Innovative Blockchain-based Farming Marketplace and Smart Contract Performance Evaluation

Guilain Leduc^{a,*}, Sylvain Kubler^a, Jean-Philippe Georges^a

^aUniversité de Lorraine, CNRS, CRAN, F-54000 Nancy, France

Abstract

E-Agriculture, or Smart Farming, refers to the design, development, and application of innovative methods to use modern information and communication technologies (ICTs), such as the Internet of Things (IoT) and machine learning, to move towards more sustainable agricultural and farming practices. The integration of blockchain technology in farming is gaining attention for its potential to migrate from the centralized and monopolistic model that shapes today's food value chain. This paper highlights the fact that most of today's blockchain-based farming frameworks focus on food tracking and traceability. Only rarely does research focus on the design of digital marketplaces to support the trading of agricultural goods between farmers and potentially interested third party stakeholders; equally rarely are performance evaluations performed for the proposed frameworks. The latter is where this paper contributes the most by, not only proposing a novel blockchain-based farming marketplace platform (called "FarMarketplace"), but also a comprehensive methodology to help software solution integrators to better understand and measure how a given configuration setting of such a platform can influence the overall quality of service performance in the long run.

Keywords: Blockchain, Trading, Smart Agriculture, Network Performance, Quality of Service

1. Introduction

Agriculture is a sector that is in constant demand. Owing to the increased global population and limited (or scarce) resources, this demand is continuously increasing, enlarging the demand-supply gap (Blandford, 2019). This clearly poses new challenges including the lack of traceability and control throughout the food supply chain, lack of quality assurance, and trust challenges resulting from the growth and consolidation of corporate monopoly power in the food industry (Zhao et al., 2019). Consequently, farmers and the food industry in general are increasingly searching for and adopting new strategies based on modern ICT such as the IoT, cloud computing, big data, and blockchain (Rabah, 2018; Lin et al., 2017; Aker et al., 2016). These emerging technologies have led to the phenomenon of e-agriculture, also referred to as Agriculture 4.0 or smart farming/

precision¹ (Lezoche et al., 2020; Wolfert et al., 2017; Vermesan and Friess, 2016), which contributes to making farms more connected, intelligent, and thus more sustainable (Krishnan et al., 2020; Kamble et al., 2019; Klerkx et al., 2019; Rose and Chilvers, 2018).

The emergence of blockchain technology, a distributed ledger technology, has raised significant expectations for moving towards more sustainable farming systems and practices at different levels of the triple bottom line: Social, Environmental, and Economic (Pinto et al., 2019; Tripoli and Schmidhuber, 2018). First, it has the power to break the centralized, monopolistic, asymmetric, and opaque model that shapes today's food value chain (Zhao et al., 2019). Secondly, blockchain offers a unique set of capabilities including decentralization, immutability, transparency, and fault-tolerance, which enables trustless architecture models that were impossible to conceive only a small number of years ago (Bermeo-Almeida et al., 2018; Feng et al., 2020; Rabah, 2018). In recent years, an increasing number of scientific and industrial blockchain-based farming initiatives

*Corresponding author

Email addresses: guilain.leduc@univ-lorraine.fr (Guilain Leduc), s.kubler@univ-lorraine.fr (Sylvain Kubler), jean-philippe.georges@univ-lorraine.fr (Jean-Philippe Georges)

¹Although one could argue that distinctions exist between these concepts, we interchangeably use these terms in the remainder of this article.

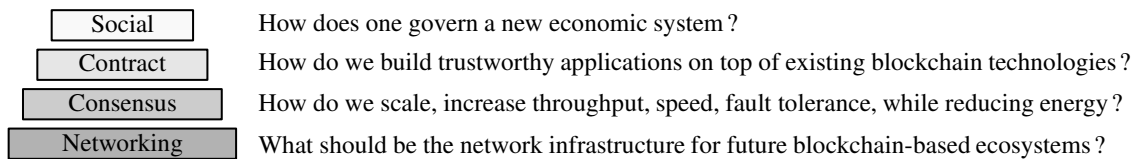


Figure 1: Blockchain Architecture Stack and associated research questions

39 have appeared around the world (Lin et al., 2017; Kami- 81
 40 laris et al., 2019), wherein different research questions 82
 41 at the hardware, software, network, and governance lev- 83
 42 els have been addressed. 84

43 Whereas the majority of the research has focused 85
 44 on investigating blockchain-based solutions for the en- 86
 45 hanced tracking and traceability of agricultural goods, 87
 46 as is further analyzed and discussed in Section 2, limited 88
 47 research has been undertaken on the design of innova- 89
 48 tive smart farming digital marketplaces to support the
 49 trading of agricultural goods between farmers and in-
 50 terested third party stakeholders (e.g., food transforma-
 51 tion companies, retailers, and other farmers). To over-
 52 come this gap in research, this paper presents a novel
 53 digital marketplace called “FarMarketplace”. FarMar-
 54 ketplace fully exploits the advantages of blockchain ca-
 55 pabilities by proposing generic, yet detailed represen-
 56 tations of trading (smart) contract templates between
 57 farmers, interested third-party consumers, and deliver-
 58 ers. Compared with the current literature, FarMarket-
 59 place is innovative in three respects.

- 60 • The evaluation of this blockchain solution facili- 91
 61 tates a methodology to benchmark a blockchain 92
 62 system. It focuses most notably on the allowable 93
 63 capacity offered by the blockchain immediately be- 94
 64 fore saturation. Hence, this evaluation is based on 95
 65 the expected contract emission throughput and its 96
 66 latency according to the block size. Consequently, 97
 67 the notion of *capacity*, the maximum throughout 98
 68 that the chain can support is introduced. Phenom- 99
 69 ena around the *capacity* are also presented, and the 100
 70 methodology is applied for a specific contract. 101
- 71 • Only limited blockchain-based farming frame- 102
 72 works/ecosystems focus on trading and propose 103
 73 any kind of digital marketplaces where farmers/in- 104
 74 dustries, deliverers, and retailers can discover each 105
 75 other and trade agricultural goods and delivery ser- 106
 76 vices; 107
- 77 • The majority of the studies do not provide suf- 108
 78 ficient details regarding the performance of their 109
 79 proposed system (more than 50% do not evaluate 110
 80 any metrics), and to the best of our knowledge, no 111

study has ever defined a comprehensive methodol-
 ogy to assist software solution integrators under-
 stand the performance characteristics and long-run
 capacity limits – *from a quality of service (QoS)*
standpoint – of FarMarketplace-like platforms.

The FarMarketplace specifications and performance
 assessment methodology are detailed in Section 3. The
 performance evaluation of FarMarketplace is presented
 in Section 4. The conclusion follows.

2. Blockchain-based Smart Farming

A brief overview of the blockchain-related back-
 ground is given in Section 2.1. In Section 2.2, past
 and ongoing blockchain-based farming/agricultural ini-
 tiatives are reviewed and analyzed. Based on this liter-
 ature review, Section 2.3 discusses the extent to which
 our research advances the current state-of-the-art.

2.1. Blockchain background and positioning

Increasing attention has been devoted to blockchain
 over the past years as it offers powerful tamper-proof
 logging and auditing capabilities where trust and control
 are no longer centralized and black-boxed, but rather
 decentralized and transparent (i.e., no requirement for a
 central trusted authority) (Zheng et al., 2018; Panarello
 et al., 2018). The possibility of defining/using “Smart
 contract” has opened a wide spectrum of applications
 where blockchain technology can be leveraged, and
 identified an entire new class of business models for
 shared data (Nowiński and Kozma, 2017). In this re-
 spect, a number of consortia are working on the de-
 sign of decentralized digital marketplaces in different
 sectors such as healthcare, logistics, energy, construc-
 tion, agriculture, and telecommunication (Al-Jaroodi
 and Mohamed, 2019). Domain-independent initiatives
 are also being identified, such as Trusted IoT Alliance²
 and IOTA Foundation³, Enterprise Ethereum Alliance
 (EEA), and Flowchain⁴. All these initiatives promote

²<https://www.trusted-iot.org>, last access Apr. 2020

³<https://www.iota.org>, last access Apr. 2020

⁴<https://flowchain.co>, last access Apr. 2020

and investigate different, yet common architectural design principles and best practices to achieve specific requirements. These challenges occur at multiple layers of the blockchain stack, as emphasized in Figure 1.

“Consensus” and “(Smart) Contract” are the most discussed layers; the former allowing the secure updating of a distributed shared state, the latter allowing the implementation of user-defined operations of arbitrary complexity that are not possible through plain cryptocurrency protocols such as bitcoin. However, the Consensus layer is undoubtedly the one that has the most influence on network performance, which is strongly dependent on the type of consensus supporting the selected/implemented blockchain technology. Consensus protocols are typically grouped into one of three categories: (i) *Permissionless (Public)*: anyone can join, transact, and review the chain without a specific identity; there is no censorship method; (ii) *Permissioned (Private)*: a type of permission is required to access all or part of the blockchain; (iii) *Federated (Consortium)*: this is a hybrid between the two previous groups. Whereas permissionless blockchains are highly scalable, fault-tolerant, and persistent, they suffer from poor performance with high latency, low throughput, and high-energy consumption. The opposite applies to permissioned blockchains. It is thus important for software solution integrators to be aware of the extent to which a given blockchain technology influences the overall application performance.

2.2. Current status of affairs of blockchain-based farming solutions

A number of blockchain-based agricultural solutions and platforms are emerging throughout the world (Juma et al., 2019), from startup developments such as Skuchain⁵, Provenance⁶, AgriDigital⁷, (Xu et al., 2019) and Farm Share⁸ to larger companies such as Cargill Risk Management (Dujak and Sajter, 2019).

Even though blockchain is used for different purposes such as minimizing unfair pricing, product origins, and reducing multinational agricultural influence in favor of more localized economies (Hang et al., 2020; Galvez et al., 2018; Thomason et al., 2018), its primary objective is to improve transparency and traceability throughout the food chain (Feng et al., 2020; Zhao et al., 2019; Tripoli and Schmidhuber, 2018). Figure 2 provides an

overview of a traditional food chain, including the contracts that are typically established between the involved parties (Feng et al., 2020; Bumblauskas et al., 2020; Kamilaris et al., 2019). These contracts include the following:

- *F2D (Farmer-to-Deliver) and I2D (Industry-to-Deliver)*: contract terms regarding, among other things, the farming or processed food environments, origin of drug variety and processed foods, fertilizing, and product distribution requirements (e.g., cold chain);
- *D2F (Deliver-to-Farmers), D2I (Deliver-to-Industry) and D2R (Deliver-to-Farmers)*: contract terms regarding product distribution including distribution warehousing, delivery, expected product recipient (retailer or industry);
- *R2D (Retailer-to-Customer)*: contract terms regarding sales time, price, and quality.

In Table 1, we review and classify the state-of-the-art studies that consider and eventually implement blockchain technology for smart farming purposes. The papers are classified based on five criteria:

1. *Objective*: we report why blockchain is used in the study (e.g., for traceability, tracking, trading). Even if traceability and tracking are sometimes used interchangeably, a difference can be made. In a tracing system, the information flow moves backwards through the supply chain (from consumers to suppliers), whereas tracking follows the information forward (from the source to end users) (Laux and Hurburgh Jr, 2012);
2. *Smart contract support and focus*: we report whether the study makes use of smart contract(s), and if so, we indicate (i) if those contracts are formalized in the corresponding paper (“F” and “N/F” in Table 1 being the respective abbreviations for “Formalized” and “Non-Formalized”) and (ii) what chain parties are involved based on the previously introduced contract taxonomy: F2D/I2D, D2F/D2I, D2R, or R2C;
3. *Platform*: we report whether the study has considered/used an “off-the-shelf” blockchain technology/platform such as Ethereum or Hyperledger;
4. *Performance*: we report whether the study has performed and detailed any performance evaluation regarding the proposed solutions, whether in terms of time execution, network latency, throughput, security, or other factors.

⁵<http://www.skuchain.com/>, last accessed May 2020.

⁶<https://www.provenance.org/>, last accessed May 2020.

⁷<https://www.agridigital.io/>, last accessed May 2020.

⁸<http://farmshare.org>, last accessed May 2020.

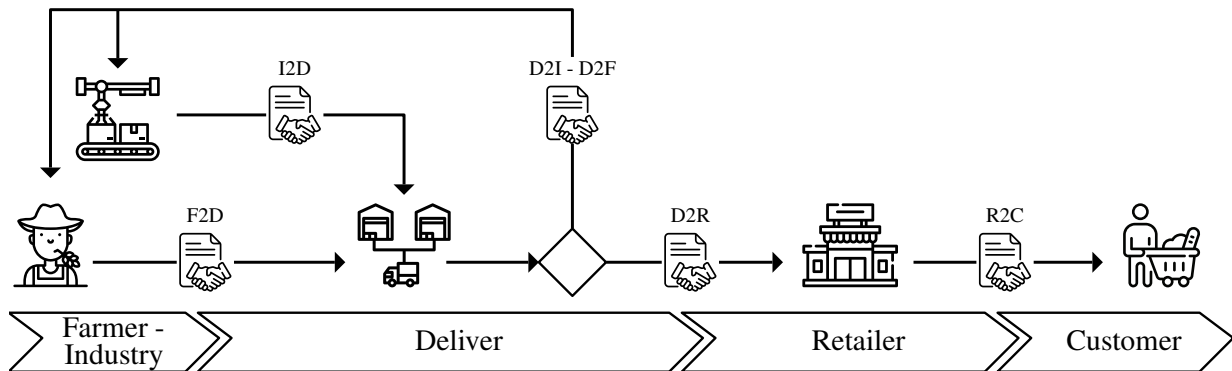


Figure 2: Traditional food chain

210 First, it can be observed that all the reported stud- 246
 211 ies have been published in the last three years, which 247
 212 confirms the growing attention paid to blockchain in 248
 213 the agricultural sector. Moreover, the majority of the 249
 214 studies (23 out of the 27 reported in Table 1) employ 250
 215 blockchain for food traceability and/or tracking pur- 251
 216 poses. The four other studies use blockchain to auto- 252
 217 mate temporary employment contracts between the 253
 218 farmers and labor contractors (Pinna and Ibba, 2018) 254
 219 and allow for agricultural resource trading between 255
 220 farmers, deliverers, and retailers (Leng et al., 2018; Mao 256
 221 et al., 2019; Bore et al., 2020).

222 Secondly, virtually all the reported studies exploit 258
 223 the smart contract capabilities to achieve the above- 259
 224 mentioned objectives (i.e., to meet traceability, track- 260
 225 ing, and trading requirements); 17 of the 27 studies fo- 261
 226 cus on – or fulfill to be more precise – market inter- 262
 227 actions between farmers/industries, deliverers, and 263
 228 retailers (i.e., F-I2D, D2F-I, R2C). Of these 17 studies, 264
 229 12 extend the traceability, tracking, or trading facilities 265
 230 to the entire food lifecycle (i.e., covering R2C inter- 266
 231 actions). It should be noted that the reported studies do not 267
 232 necessarily track/trace the same food system features. 268
 233 Indeed, certain studies such as (Hang et al., 2020; Bum- 269
 234 blauskas et al., 2020; Devi et al., 2019; Surasak et al., 270
 235 2019; Lin et al., 2018) track the environmental back- 271
 236 ground information of a food item using sensor-like de- 272
 237 vices (e.g., amount of pesticides used, temperature evo- 273
 238 lution), whereas other studies track other supply chain 274
 239 information such as (i) incident details throughout the 275
 240 crop harvesting process (Iqbal and Butt, 2020), (ii) car- 276
 241 bon footprint at food production and transportation lev- 277
 242 els (Shakhbulatov et al., 2019), and (iii) food quality 278
 243 evolution (Carbone et al., 2018; George et al., 2019).

244 Thirdly, reviewing what blockchain technologies 280
 245 have been considered in the reported studies (see col- 281

umn “Platform” in Table 1), Ethereum and Hyperledger 246
 Fabric are the most widely adopted solutions (the for- 247
 mer being used in five studies, the latter in five). This 248
 is not surprising as they are both market share lead- 249
 ers (50% of the implemented projects being hosted on 250
 these platforms) (Udokuwu et al., 2018). However, in- 251
 terestingly, one could question why studies aiming to 252
 achieve similar goals opt for one or the other? Indeed, 253
 whereas Ethereum is more suitable for permissionless 254
 distributed ledgers, Hyperledger is more suited to per- 255
 mitted blockchains (Xie et al., 2019; Sajana et al., 256
 2018). To answer this question, a more in-depth analy- 257
 sis of these studies should be performed to identify the 258
 exact system requirements and constraints.

260 Finally, it can be observed in Table 1 (see col- 261
 umn “Performance”) that less than half of the reviewed 262
 studies performed experimental evaluations of their so- 263
 lutions. In our opinion, this clearly indicates that 264
 blockchain-based farming remains in its infancy, where 265
 the focus is more on architectural and functional design 266
 choices than on performance benchmarking. For studies 267
 evaluating the performance of their solution, through- 268
 put and latency are the most used metrics, considered 269
 in 65% and 50%, respectively, of the reviewed litera- 270
 ture). Throughput corresponds to the number of suc- 271
 cessful transactions per second (a transaction being suc- 272
 cessful if it has been validated and committed to a new 273
 block); latency corresponds to the delay between the 274
 emission of a transaction and its commitment to a new 275
 block. It can be observed that only a small number of 276
 studies evaluated security aspects. The main reason for 277
 this is that the majority of the proposed solutions rely on 278
 off-the-shelf blockchain solutions, whose security per- 279
 formance – which is characterized by the number of 280
 trusted participants required to secure the blockchain – 281
 has been widely studied and described in the literature

Table 1: Current state of affairs of Smart Farming initiatives

Reference	Objective	Smart Contract (SC) support & focus					Platform	Performance
		SCs	(F-I)2D	D2(F-I)	D2R	R2C		
(Pinna and Ibba, 2018)	Temp employ.	F	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ethereum	-
(Devi et al., 2019)	Track	N/F	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ethereum	1. Latency
(Patil et al., 2017)	Track	-	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N/S	-
(Tse et al., 2017)	Trace	-	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N/S	-
(Carbone et al., 2018)	Track	N/F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Hyperledger	-
(Hang et al., 2020)	Track	F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Hyperledger (v1.4.3)	1. Throughput 2. Latency
(Lin et al., 2017)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	-	-
(Lucena et al., 2018)	Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Hyperledger	-
(Mao et al., 2019)	Trade	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Own (FTSCON)	1. Exec. time 2. Merchant profit, 3. Security
(Tian, 2017)	Trace	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ethereum	-
(Bore et al., 2020)	Trade	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hyperledger	1. Throughput 2. Latency
(Stefanova and Salampasis, 2019)	Trace	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Hyperledger	-
(Leng et al., 2018)	Trade	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	N/S	1. Throughput 2. Latency
(Kumar and Iyengar, 2017)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	-
(Iqbal and Butt, 2020)	Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-	1. ZigBee-related
(George et al., 2019)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
(Caro et al., 2018)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ethereum & Hyperledger	1. Throughput 2. Latency 3. CPU
(Surasak et al., 2019)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	SQL-based	-
(Bumblauskas et al., 2020)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Hyperledger	-
(Malik et al., 2018)	Trace	F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Hyperledger	1. Time
(Hua et al., 2018)	Trace & Track	F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/S	-
(Shakhbulatov et al., 2019)	Track	F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Raft-like consensus	1. Throughput 2. Time
(Lin et al., 2018)	Trace	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
(Reddy and Kumar, 2020)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
(Xie et al., 2017)	Track	F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ethereum (v1.9)	1. Throughput
(Papa, 2017)	Trace	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-
(Awan et al., 2019)	Trace & Track	N/F	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/S	1. Throughput

(Ali et al., 2018; Xiao et al., 2020). In fact, the security level of a given blockchain technology is directly derived from the consensus protocol supporting the chain. For example, in proof-of-work (PoW) consensus, the number of honest miners must be greater than 51%; this number must be $\geq 66\%$ in Byzantine fault tolerance (BFT) consensus algorithms Vukolić (2015). As a general remark, the current literature does not address sufficient attention to properly analyzing the extent to which a given architectural design choice can influence the long-run capacity limits (in transactions/second Tx/s) of the proposed system, and hence by definition, on the overall (end-to-end) QoS. As an example, latency is directly dependent on the throughput, and can be negatively influenced by a high delay of transaction propagation. Furthermore, if the throughput is less than the transaction asking rate, congestion is likely to occur,

which results in an increase in latency. Such interactions between blockchain- and infrastructure-related parameters are rarely analyzed and considered in the literature, thus requiring further research.

2.3. Positioning and Contribution of this research work

Based on the literature review presented in the previous section, we stress three important facts.

1. The vast majority of the studies are dedicated to traceability and tracking along supply chains, and conversely, only a limited number focus on trading, i.e., digital marketplaces where farmers/industries, deliverers, and retailers can discover each other and trade agricultural goods and delivery services.
2. The vast majority of the studies use on-the-shelf blockchain technologies; in particular, Ethereum or Hyperledger Fabric.

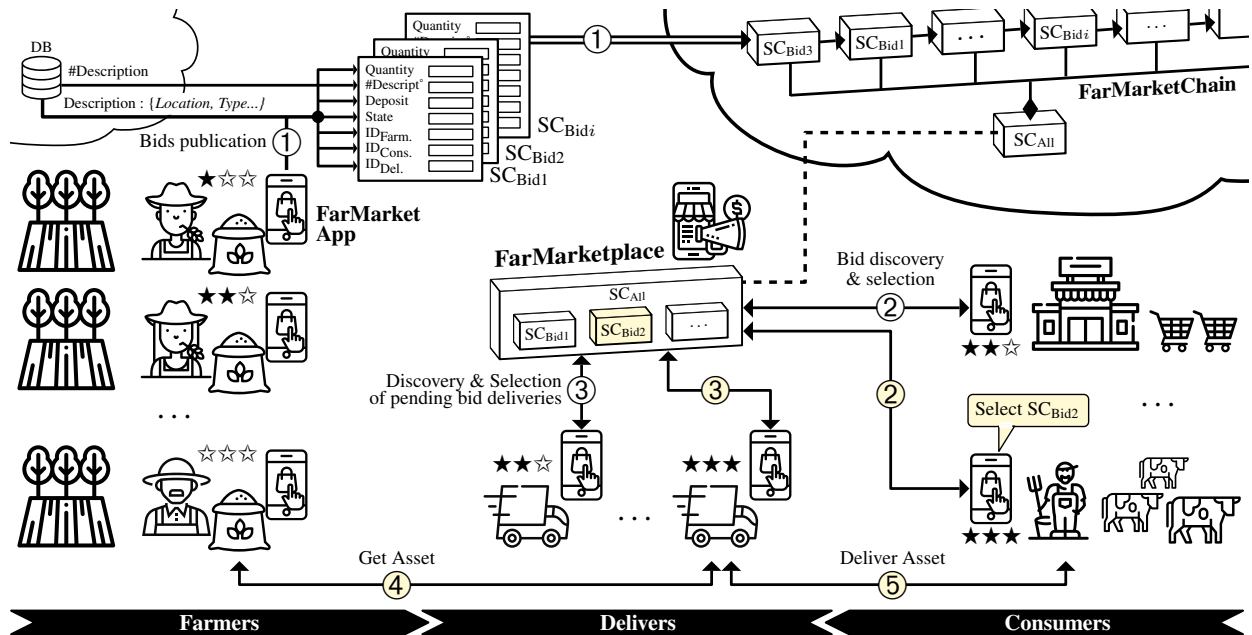


Figure 3: Overview of the FarMarket ecosystem and associated interactions between stakeholders and supporting marketplace

315 3. The vast majority of the studies do not provide 339
 316 sufficient details regarding the performance of the 340
 317 proposed system (more than 50% of the studies do 341
 318 not evaluate any metrics). This lack of compre- 342
 319 hensive evaluation in the studies, combined with 343
 320 the lack of details regarding the implemented smart 344
 321 contracts (only 20% of the studies provide rele- 345
 322 vant details) and lack of details regarding the 346
 323 blockchain configuration, which has a direct im- 347
 324 pact on the overall system performance, makes 348
 325 it difficult to compare existing blockchain-based 349
 326 farming frameworks.

327 Given the above findings, this paper advances the cur- 353
 328 rent state-of-the-art in two respects. First, a novel digital 354
 329 marketplace for agricultural product trading purposes is 355
 330 proposed, which is in agreement with research by Mao 356
 331 et al. (2019); Bore et al. (2020); Leng et al. (2018). Sec- 357
 332 ondly, a representation of the interactions that occur be- 358
 333 tween blockchain- and/or infrastructure-related param- 359
 334 eters is presented. This representation not only provides 360
 335 software solution integrators with a holistic overview of 361
 336 possible interactions, but also facilitates the analysis of 362
 337 the system (QoS) performance limitations in the long 363
 338 run.

3. FarMarket ecosystem

The digital FarMarketplace proposed in this study is a part of a larger ecosystem referred to as “FarMarket”. The building blocks supporting this ecosystem are presented in Section 3.1 and the smart contract templates specified for trading are further detailed in Section 3.2. In Section 3.3, key performance indicators in the performance evaluation process of a FarMarket-like ecosystem are discussed.

3.1. Ecosystem services and supporting architecture

An overview of the different stakeholders and software/hardware components supporting the FarMarket ecosystem is depicted in Figure 3. This ecosystem is designed to:

- collect agricultural bids/contracts published by farmers or other like-minded providers (see ① in Figure 3);
- notify consumers that new bids/contracts are available, allowing them to select/purchase one or more contracts/bids (see ②);
- notify deliverers that new delivery offers are available, allowing them to select one or more offers (see ③), which – if accepted by the farmer and consumer – implies that the deliverer must collect and

363 deliver the asset associated to the bid/contract (see
364 ④ and ⑤), upon which they will be paid;

- 365 • allow all stakeholders to evaluate the service qual-
366 ity, namely (i) the consumer can evaluate the qual-
367 ity of the delivery service (e.g., punctuality, profes-
368 sionalism) and the received agricultural goods, (ii)
369 the deliverer can evaluate the quality of the farmer
370 and consumer (e.g., punctuality, accuracy of the
371 specified location); and (iii) the farmer can eval-
372 uate the quality of the delivery service.

373 To achieve the above functionalities, three main
374 building blocks have been designed and integrated into
375 the FarMarket ecosystem.

- 376 1. *FarMarketchain*: This refers to the blockchain and
377 associated smart contracts. A database, denoted by
378 DB in Figure 3, functions with blockchain to avoid
379 storing long chains of characters in the blockchain
380 itself, which is costly (only the hash of the corre-
381 sponding chain is added to the blockchain).
- 382 2. *FarMarketplace*: This refers to the digital market-
383 place platform. It hosts the blockchain and DB,
384 and has the role of intermediary between the dif-
385 ferent ecosystem stakeholders.
- 386 3. *FarMarketApp*: This refers to the App that allows
387 stakeholders to benefit from the set of services of-
388 fered by the FarMarket ecosystem.

389 Using an exterior database to store data is a common
390 practice in blockchain development. Indeed, it may be-
391 come expensive to store raw information in a distributed
392 ledger, as each transaction usually implies a fee, and
393 storing only the hash of information stored in a database
394 (allowing for verifying the data integrity by comparing
395 that hash at any given time) is a widely adopted alter-
396 native. This database is, in our case, a server but can
397 be substituted by a private cloud [Sumathi et al \(2020\)](#) or
398 IPFS (InterPlanetary File System) to allow for fully de-
399 centralized peer-to-peer framework [Singh et al \(2019\)](#).
400 As the access control of data is not the main focus of
401 this paper, all data is freely accessible, although more
402 advanced access control strategies could be adopted in
403 the future, as defining a XACML politic [Ramli et al
404 \(2014\)](#) or adopting a blockchain-based solution [Maesa
405 et al \(2017\); Esposito et al \(2021\)](#).

406 3.2. Smart farming contracts

407 The set of interactions (or communications) occur-
408 ring between the previously introduced building blocks

Algorithm 1: create_SC_{Bidi}

```

Input : IDUser, Desc
1 SCBidiIDFar ← IDUser; // Initialize contract's owner ID
2 SCBidi#Desc ← hash(Desc); // Compute #Desc value
3 SCBidiPFar ← priceBid(Descquant, Desctype...); // Compute bid
   price
4 SCBidibalance ← 0; // Initialize bid's balance
5 SCBidistate ← Available; // Initialize contract state
Output: SCBidi

```

Algorithm 2: purchase_SC_{Bidi}

```

Input : locIDCon, IDUser, Desc, amount, PDel
1 if SCBidistate == Available & amount == (SCBidiPFar + PDel) then
2   SCBidiIDCon ← IDUser; // Update consumer's location
3   SCBidiPDel ← PDel; // Set service delivery price
4   Desc ← Desc ∪ {locIDCon}; // Update DB description
5   SCBidi#Desc ← hash(Desc); // Compute new #Desc value
6   SCBidibalance ← amount; // Deposit
7   IDConbalance ← IDConbalance - amount; // Update balance
8   SCBidistate ← WaitForDeliverer; // Update SC state
Output: SCBidi

```

and stakeholders are further detailed in Figure 4 in the
form of a sequence diagram.

First, consumers and deliverers can look for bids and
pending delivery offers (*cf.*, ① in Figure 4), and possi-
bly subscribe to the FarMarketplace platform to be noti-
fied whenever a new bid/offer is published (the Message
Queuing Telemetry Transport protocol is being used in
this respect). Farmers can publish new bids by speci-
fying – *via the FarMarketApp* – information related to
their bid (*cf.*, ②). This action calls the bid creation func-
tion detailed in Algorithm 1, where SC_{Bidi} refers to a
given smart contract (*i* referring to the *i*th contract). In
fact, two input parameters are sent by the *FarMarke-*
tApp to Algorithm 1, namely, (i) *Farmer's ID* and (ii)
Description (consisting of several information items as
detailed in Table 2). Based on these two input param-
eters, five immutable attributes – *also referred to as*
“state variables” – are extracted/derived to be stored in
the *FarMarketchain*, namely: (i) Farmer's ID, (ii) *hash*
value of the bid description, (iii) bid price, (iv) contract
balance, and (v) contract state. The input parameters
communicated by *FarMarketApp* and derived state vari-
ables are summarized in Table 2.

At this stage, the contract is available on the market-
place and consumers have been notified of its existence
(③). When a consumer selects a bid for purchase (*cf.*,
④), the purchase function of SC_{Bidi} is executed, as de-
tailed in Algorithm 2. The purpose of this function/al-
gorithm is to verify that the contract is in the correct
state (should be *Available* for purchase) and that the

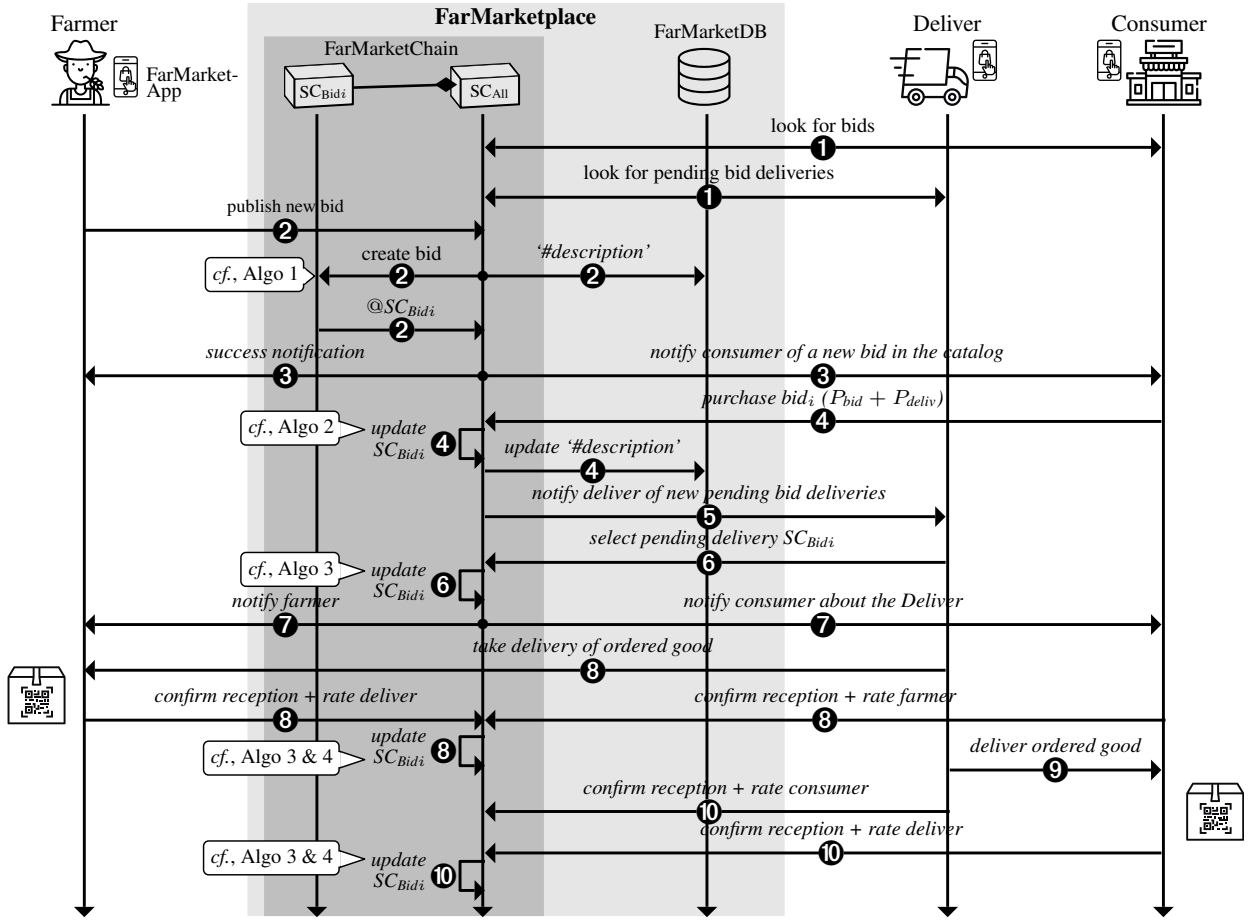


Figure 4: Messaging protocol supporting FarMarket ecosystem

439 consumer has sufficient money in her/his digital wallet. 454
 440 The amount of money required to purchase the contract 455
 441 should be equal to the bid price plus the service delivery 456
 442 price, which is denoted by P_{Del} . Note that P_{Del} is an 457
 443 input of the purchase function, meaning that it is computed 458
 444 outside the smart contract⁹ and then added to the 459
 445 corresponding state variable (cf., line 3 of Algorithm 2). 460
 446 If these conditions are satisfied, the following state vari- 461
 447 ables are updated: (i) contract's subscriber/beneficiary 462
 448 with the consumer's ID; (ii) contract delivery price; (iii) 463
 449 consumer's location, which is part of the bid descrip- 464
 450 tion; (iv) contract balance credited with the required 465
 451 amount; and (v) contract state set to *WaitForDeliverer*. 466
 452 In addition, the balance of the consumer's wallet is up- 467
 453 dated accordingly (cf., line 7 in Algorithm 2). Once the

⁹The function for computing the service delivery price is beyond the scope of this paper; however, it can be computed on the basis of parameters such as the consumer's location and type of goods to be delivered

454 contract state has been updated to *WaitForDeliverer*, del- 455
 456 liverers who have subscribed to pending bid delivery of- 457
 458 fers are notified (cf., ⑤).

457 When a deliverer selects a pending bid delivery offer 458
 459 (cf., ⑥), the delivery function of SC_{Bid_i} is executed, as 460
 461 detailed in Algorithm 3. This function/algorithm ver- 462
 463 ifies the contract state. If state is *WaitForDeliverer*, 464
 465 then the deliverer becomes the official delivery service 466
 467 provider (cf., line 2 in Algorithm 3), the contract state 468
 469 becomes *WaitForDelivery*, and both the consumer and 470
 471 farmer are notified of the deliverer's identity (cf., ⑦). 472
 473 At this stage, the deliverer must take delivery of the or- 474
 475 dered goods (cf., ⑧). At the moment of exchanging the 476
 477 goods, the deliverer and farmer must both confirm the 478
 479 successful reception, which in practice, results in the 479
 480 call of the *delivery_SC_{Bid_i}* function, leading to a change 480
 481 in the contract state to *OnDelivering* (cf., lines 4-5 in 481
 482 Algorithm 3). In the final stage, the deliverer delivers 482
 483 the goods to the consumer (cf., ⑨). Both confirm the

Table 2: State and Input variables related to FarMarket Application

	Variable	Description
App Inputs	ID_{User}	User identifier referring either to a farmer, deliverer, or consumer, respectively denoted by ID_{Far} , ID_{Del} , or ID_{Con}
	$ID_{User}^{balance}$	Balance of user's wallet
	Amount	Amount paid by a consumer to purchase a given contract/bid denoted by SC_{Bidi}
	P_{Far} , P_{Del}	Prices of (i) agricultural goods to be paid to the farmer computed using the <i>priceBid()</i> function, which takes as inputs: product type and quantity and (ii) service delivery that depends, among other inputs, on the distance between consumer and farmer
	Desc	Description stored in the database (see DB in Figures 3 and 4), consisting of: (i) farmer's and consumer's location denoted by $loc^{ID_{Far}}$ and $loc^{ID_{Con}}$, respectively; (ii) type; (iii) quantity of the agricultural goods; and (iv) additional comments
	$R_{User1 \rightarrow User2}$	Rating score referring to how satisfied <i>User1</i> is regarding the 'service' delivered by (or the behavior of) <i>User2</i> . All possible rating score combinations are contained in a set denoted by $\mathcal{R} = \{R_{Far \rightarrow Del}, R_{Del \rightarrow Far}, R_{Del \rightarrow Con}, R_{Con \rightarrow Far}, R_{Con \rightarrow Del}\}$
State Variables	SC_{Bidi}^{state}	State variable referring to the state of contract SC_{Bidi} at a given point in time. Possible states are <i>{Available, WaitForDeliverer, WaitForDelivery, OnDelivery, Delivered}</i>
	$SC_{Bidi}^{ID_{User}}$	State variable referring to a given stakeholder (<i>cf.</i> , ID_{User})
	SC_{Bidi}^{Desc}	State variable referring to the hash of the bid's description (<i>cf.</i> , "Desc"), obtained using the <i>priceBid(...)</i> function
	$SC_{Bidi}^{balance}$	State variable referring to the balance of the smart contract/bid <i>i</i>
	$SC_{Bidi}^{P_{Far}}$, $SC_{Bidi}^{P_{Del}}$	State variables referring to prices to be paid to the farmer and deliverer (<i>cf.</i> , P_{Far} , P_{Del})
	$SC_{Bidi}^{R_{User1 \rightarrow User2}}$	State variable referring to the satisfaction rating scores previously described (<i>cf.</i> , \mathcal{R})

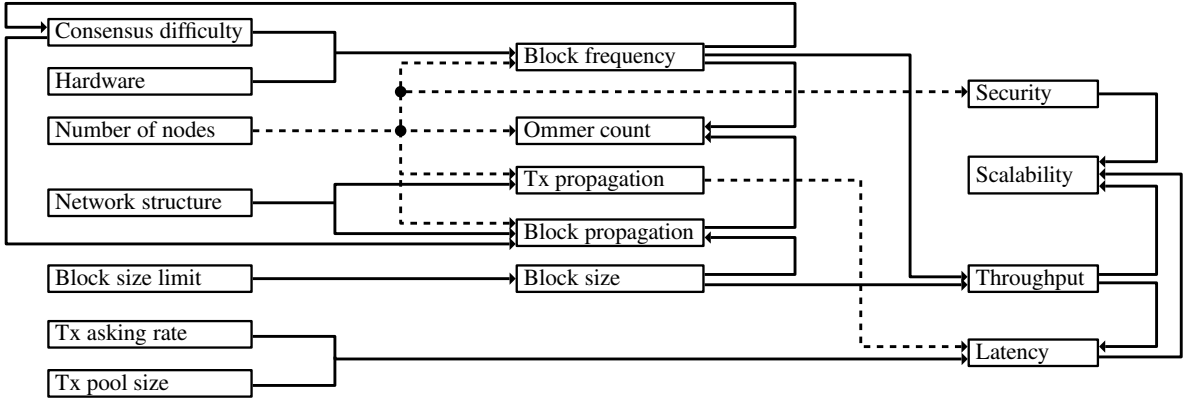


Figure 5: Performance interaction models regarding blockchain- and infrastructure-related parameters (note, that there is no difference between the solid and dashed arrows, they are used for figure clarity only)

473 successful delivery/reception (*cf.*, ⑩), which leads to a 489
 474 change in the contract state to *Delivered* (*cf.*, lines 6-7 490
 475 in Algorithm 3), following which payments are made to 491
 476 the farmer and deliverer, and the balances updated ac- 492
 477 cordingly (*cf.*, lines 8-10). 493

478 To provide stakeholders with the possibility of evalu- 495
 479 ating service quality, as previously discussed in Section 496
 480 3.1, another function is defined in the smart contract to 497
 481 make satisfaction scores immutable. This function is 498
 482 detailed in Algorithm 4, allowing consumers, farmers, 499
 483 and deliverers to evaluate each other through a rating 500
 484 score denoted by $R_{User1 \rightarrow User2}$ (*cf.* Table 2). These rating 501
 485 scores refer to the reputation/satisfaction level related 502
 486 to a given FarMarketplace's stakeholder. Note that the 503
 487 functions used for computing these rating scores are not 504
 488 included in the scope of this paper. 505

3.3. Ecosystem-related Key Performance Indicators

As discussed in Section 2, only limited interactions between blockchain- and infrastructure-related parameters are formalized in the literature, although it is essential to have a comprehensive understanding of such interactions to address the QoS requirements. It is clearly not that simple to develop a unique model/representation of such interactions, as there could be as many models as there are blockchain technologies (e.g., because of different consensus mechanisms). In this section, we attempt to clarify, in a graphical manner in Figure 5, the parameter interactions of PoW-based blockchain technologies. The parameters listed on the left side of the figure correspond to application-specific parameters (e.g., implemented network architecture, number of nodes/users), whereas the parameters in the center of the figure refer to features that are specific/intrinsic

Algorithm 3: delivery_SC_{Bidi}

```

Input : IDUser
1 if SCBidistate == WaitForDeliverer then
2   | SCBidiIDDel ← IDUser; // Set deliverer's contract
3   | SCBidistate ← WaitForDelivery; // Update SC state
4 else if SCBidistate == WaitForDelivery & SCBidiIDDel == IDUser
   then
5   | SCBidistate ← OnDelivering; // Update SC state
6 else if SCBidistate == OnDelivering & SCBidiIDCon == IDUser then
7   | SCBidistate ← Delivered; // Update SC state
8   | IDFarmbalance ← IDFarmbalance + SCBidiPbid; // Update farmer
   | balance
9   | IDDelbalance ← IDDelbalance + SCBidiPdel; // Update deliverer
   | balance
10  | SCBidibalance ← 0; // Set SC balance to zero
Output: SCBidi

```

Algorithm 4: rating_SC_{Bidi}

```

Input : IDUser,  $\mathcal{R}$ ; // Consumer location
1 if SCBidistate == Delivered & IDUser == SCBidiIDFar then
2   | SCBidiRFar→Del ←  $\mathcal{R}$ Far→Del; // Set rating score
3 else if SCBidistate == Delivered & IDUser == SCBidiIDDel then
4   | SCBidiRDel→Far ←  $\mathcal{R}$ Del→Far; // Set rating score
5   | SCBidiRDel→Con ←  $\mathcal{R}$ Del→Con; // Set rating score
6 else if SCBidistate == Delivered & IDUser == SCBidiIDCon then
7   | SCBidiRCon→Far ←  $\mathcal{R}$ Con→Far; // Set rating score
8   | SCBidiRCon→Del ←  $\mathcal{R}$ Con→Del; // Set rating score
Output: SCBidi

```

to the implemented blockchain technology (i.e., non-configurable parameters); the parameters on the right side refer to QoS performance metrics. An arrow from a frame A to a frame B indicates that parameter A has an influence, to a greater or lesser degree, on parameter B (or performance metric B). The following discusses the identified interactions.

First, the “consensus difficulty” lies in the complexity of generating a block in the chain. It is known in the literature that the time required to solve this challenge is linked to the computational power of the network, which is composed of the “number of computational nodes” and associated “hardware” resources (e.g., allocated threads, memory, processors) (Pierro, 2019). These three parameters (consensus difficulty, hardware, number of nodes) inevitably influence the “Block (generation) frequency” parameter, as emphasized in Figure 5.

The “Network structure”, which includes data distribution mechanisms, regroups parameters that influence the delay for broadcasting transactions and blocks in the chain. High block propagation delays, associated with

high block generation frequency, increase the likelihood of forming concurrent blocks in the blockchain network nodes. Such concurrent blocks, which are called “ommer” (or “uncle”) blocks in Ethereum, could possibly not be included in the main chain.

The “number of nodes” in the blockchain network has an influence on the overall system performance as it influences the block and transaction propagation process. Indeed, the greater the number of nodes, the greater the number of messages to be propagated over the network. Another important interaction to be aware of is between the “number of nodes” and “security”, as the greater the number of nodes, the greater the level of security. This interaction applies not only to PoW-based blockchain technologies, but also to technologies using BFT-like consensus.

The “Block size” parameter, which is limited by the “Block size limit” set at the configuration stage (e.g., gas limit in Ethereum), has a direct influence on the “block propagation” parameter, as well as on the system “throughput” performance. As highlighted in Figure 5, throughput is tightly coupled with the block generation frequency and block size parameters, as the product of both results in the memory throughput where transactions are written.

The latency in blockchain networks is directly dependent on the throughput parameter, although it can be negatively influenced by high transaction propagation delays. Furthermore, as discussed in Section 2.2, if the throughput is less than the transaction asking rates, congestion effects occur, resulting in an increase in latency.

Given the above discussion, we claim in this paper that it is of the utmost importance to evaluate what level of performance a given blockchain-based system, such as the proposed FarMarket ecosystem, can achieve/support in the long run. In this study, we are particularly interested in identifying the maximum achievable throughput when the blockchain is in a steady state and not saturated (which would inevitably contribute to an increase in latency in such cases). This throughput limit is referred to as the (*long-run*) capacity in this study and is experimentally studied in the next section.

4. Implementation and Performance Evaluation

The FarMarket ecosystem and associated building blocks (i.e., FarMarketchain, FarMarketplace, FarMarketApp) were implemented for experimental and evaluation purposes. The Ethereum platform was used for

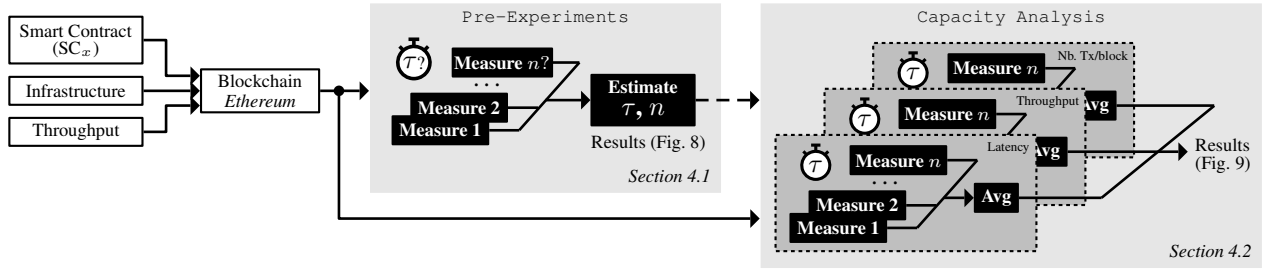


Figure 6: Methodology for fixing design of experiments

integrating the set of smart contracts¹⁰.

An overview of the experimental methodology is displayed in Figure 6. A pre-experiment stage was performed to analyze and estimate the experimental settings including the appropriate number and duration of experiments to be performed. A second stage was then performed to experimentally evaluate the QoS offered by the overall ecosystem, with a focus on latency and throughput performance metrics, in addition to the number of transactions per block. These two stages are presented in Sections 4.1 and 4.2, respectively. The experimental results are further analyzed and discussed in Section 4.3, underlining the relation between the maximal throughput offered by the chain (i.e., the long-run capacity limit) and block size.

4.1. Selection and configuration of the benchmark

In Ethereum, different feedback controllers are implemented to balance the security/robustness (related to the computational cost) of the blockchain and QoS – mainly in terms of throughput and latency – offered to support smart contracts. In fact, the hashing power directly influences the time to resolve a block, i.e., the delay of mining a block, which by definition, influences the latency. In this respect, in Ethereum, the difficulty in mining blocks (a statistical estimate of the number of hashes that must be generated to find a valid solution) is re-targeted over time to control this mining delay.

Figure 7 highlights the long-term evolution (over a 12 hour period) of the difficulty in our setup. It can be observed that the difficulty is frequently re-targeted; however, it tends to converge towards an asymptote. In fact, the system responds by increasing/decreasing the difficulty if the previous blocks are generated faster/slower than a specified mining block time, which ranges from 9 to 17 seconds (Pierro, 2019).

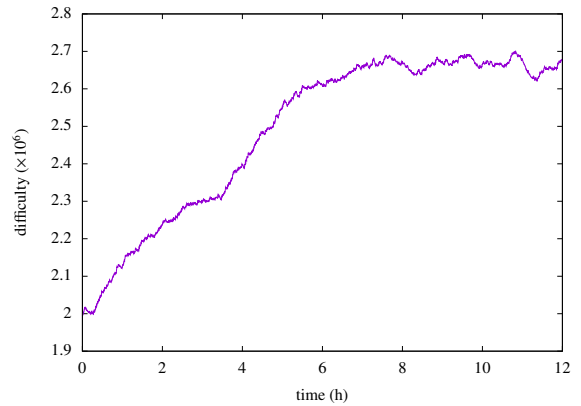


Figure 7: (Mining) difficulty per timestamp blocks

Compared with other state-of-the-art research works, our performance evaluation experiments focus on long-run QoS performance, i.e., when QoS no longer varies because of feedback control. To accelerate the control and avoid response time issues, the initial difficulty of the blockchain genesis block is set directly to its long-run value at the steady state. This value corresponds to $10 \times \#_{total}$, where $\#_{total}$ corresponds to the sum of each hashrate (the number of hashes realized by a node every second) of the computers in the network. Compared with other studies, this also allows us to focus on the real capacity of the chain and to mitigate the difference in hardware resources (e.g., number of threads, memory allowed, processors). This focus corresponds to the implementation of the proposed smart contract in Solidity (v0.5.1) on a chain shared with three computers running with Geth 1.9.6 on Ubuntu 18. The three nodes are defined as miners, one being responsible for generating the transactions. Because the latency and throughput are both influenced by block propagation delays (cf., Figure 5), the nodes are connected over a switch offering a high bandwidth to maintain delays (which are non-controllable) to a minimum, and thus allow experiments

¹⁰Solidity codes of the contracts are publicly available at the following URL: <https://github.com/inprenable/FarMarketplace>, last access Apr. 2020

633 to be reproducible. To complete the genesis block, the 685
634 gas limit, which as we know influences the block size, 686
635 is set arbitrarily to 16 777 216 gas. Furthermore, to 687
636 ensure that our experimental platform would demonstrate 688
637 the expected behavior in terms of difficulty ($\#_{total}$) and 689
638 the influence of the number of threads, a preliminary 690
639 experimental analysis was performed, as presented in 691
640 [Appendix A](#). 692

641 Based on this configuration setting, pre-experiments 693
642 were performed to determine the number (n) and du-
643 ration (τ) of the experiments to be reproduced in the 694
644 second evaluation stage. As mentioned previously,
645 the latency and throughput offered by the core chain
646 were considered the performance metrics (both being
647 obtained by comparing the timestamps when transac-
648 tions were generated for a contract and submitted to the
649 chain). Figure 8 displays the evolution of both met-
650 rics, in addition to the number of transactions per block.
651 Note that the experiments were performed after a 30 min
652 block generation period to ensure that stability of the
653 difficulty was achieved.

654 Figure 8(a) provides insight into the evolution of the
655 number of transactions per block for ten minutes, for ten
656 experiments, one transaction being submitted per sec-
657 ond. It can be observed that the convergence is rela-
658 tively fast and the blockchain remains (reasonably) sta-
659 ble. This is also confirmed by observing the QoS met-
660 rics, namely the latency (see Figure 8(e)) and through-
661 put (see Figure 8(c)). Throughput is computed as the
662 number of transactions for a block divided by the de-
663 lay to mine that block; latency is based on the time
664 difference between the emission and validation of the
665 transaction. After ten minutes, the latency is in the
666 expected range defined earlier (with a standard deviation
667 less than one second and an average mining delay of
668 11.6 seconds per block). These experiments allowed us
669 to select a simulation duration of $\tau = 10$ min for the
670 second experimental evaluation stage (a sufficient num-
671 ber of samples being available, 52 blocks on average).
672 In a further step, the confidence was analyzed by con-
673 sidering a greater number of experiments.

674 Figure 8(b) displays the evolution of the average
675 number of transactions per block when experiments
676 were added. The relative error corresponds to the dif-
677 ference between the average for a given number and av-
678 erage for 30 experiments. Figures 8(d) and 8(f) provide
679 insight into the same analysis for the throughput and la-
680 tency metrics, respectively. It can be observed that the
681 steady state is achieved with ten experiments and that
682 the increase in the number of experiments does not sig-
683 nificantly change the precision. Consequently, in the
684 following, each configuration is repeated $n = 10$ times.

Because our main goal is to evaluate the entire
ecosystem, the following section aims to define the
maximal service that can be offered to support the emis-
sion of smart contracts. It is, then, important to iden-
tify the capacity offered by the chain to store contract-
related information. In this respect, the experiments are
repeated not only according to the parameters identified
in this section, but also by increasing the transaction-
emission rate.

4.2. Analysis of the FarMarketChain Capacity

Figure 9 displays the evolution of the three metrics
previously identified; however, this time for a given
transaction-emission rate ranging from 1 Tx/s (as pre-
vious) to 21 Tx/s. Each point consists of experiments of
ten minutes, repeated ten times. As the `gasLimit` was
arbitrarily chosen (`1 blockSize = 16 777 216 gas`), we
repeated the same set of experiments on another chain
with a ten times greater `gasLimit`. Figures 9(d), 9(b),
and 9(f) correspond to this second experiment.

4.2.1. Number of transactions per block

Figures 9(a) and 9(b) provide insight into the evolu-
tion of the number of transactions inside a block ac-
cording to the transactions emission rate. Two behav-
iors emerge before and after the emission throughput
value that we refer to as *capacity*. Before the *capacity*
is achieved, the evolution of the number of transactions
per block is linear and corresponds to the average delay
of mining a block (it varies, yet is experimentally near
11 seconds) multiplied by the emission rate. Once the
capacity is surpassed, the delay required to fill a block is
less than the delay of mining a block. This results in the
filling of the blocks with the maximum number of trans-
actions, leading to an approximately constant number of
transactions per block, with minimal variation between
the two experiments.

4.2.2. Latency

Figures 9(e) and 9(f) display the average latency of
the transactions according to the throughput. We can
again extract two behaviors: before and after the *capac-
ity*. With an emission rate less than the *capacity*, as
the block is not filled, the transaction is validated and in-
serted into the next block when it reaches a node. The
latency is thus equal to the time required to wait for the
next block, which corresponds to the mining delay. This
delay differs minimally between the two experiments
(owing to the feedback control of the difficulty); hence,
the latency differs only marginally. After the *capacity*,
the system is overloaded. As the validation throughput

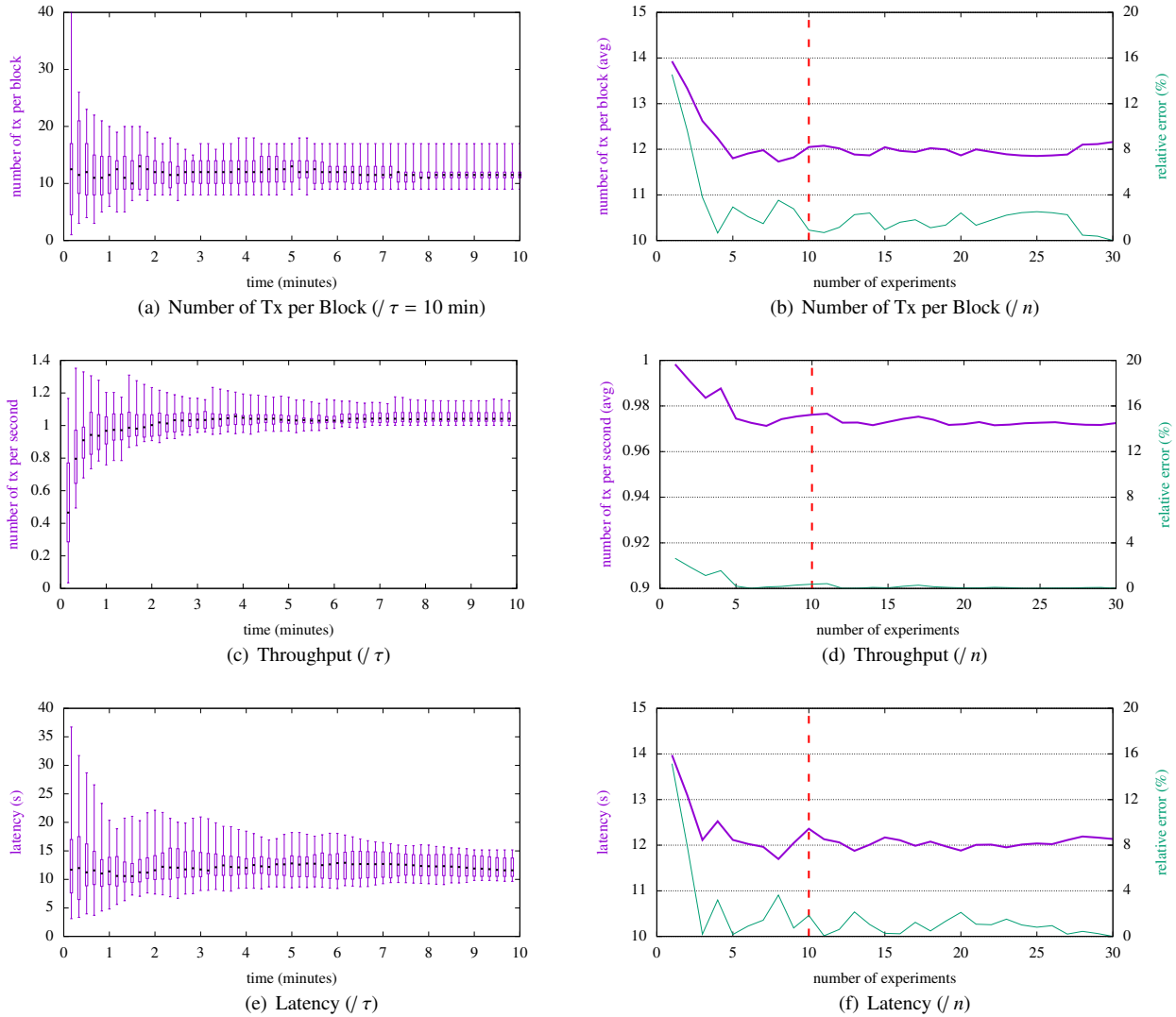


Figure 8: Evolution of QoS metrics for different number of experiments (n) and in time (τ)

733 is less than the emission rate, the transaction is queued,
734 leading to a significant increase in the latency.

735 4.2.3. Throughput

736 Figures 9(c) and 9(d) provide insight into the evolu-
737 tion of the validation throughput. In the first part, the
738 validated throughput is equal to the emission rate be-
739 cause every emitted transaction is validated (leading to
740 the identity function, with minimal differences between
741 the two experiments). In the second part, blocks are sat-
742 urated such that the validated throughput is limited to a
743 constant, which corresponds to the *capacity* and is de-
744 fined by the ratio between the number of transactions
745 per block and the delay of mining a block.

746 As assumed, the number of transactions contained in

747 a block is proportional to the size of the block. By in-
748 creasing the block size by a factor of 10, we also in-
749 crease the number of transactions in a block by a factor
750 of 10. The maximal validated throughput, i.e., the *ca-*
751 *capacity*, is also proportional to the block size.

752 4.3. Discussions

753 As evidenced through the review of the literature on
754 blockchain-based e-agriculture solutions presented in
755 Section 1, a large number of the research studies did
756 not provide performance evaluation results for their so-
757 lutions, and even fewer compared their solutions with
758 other state-of-the-art approaches. In this respect, we
759 propose to compare our proposal to another smart farm-
760 ing contract, the one proposed by Tian (2017). This con-

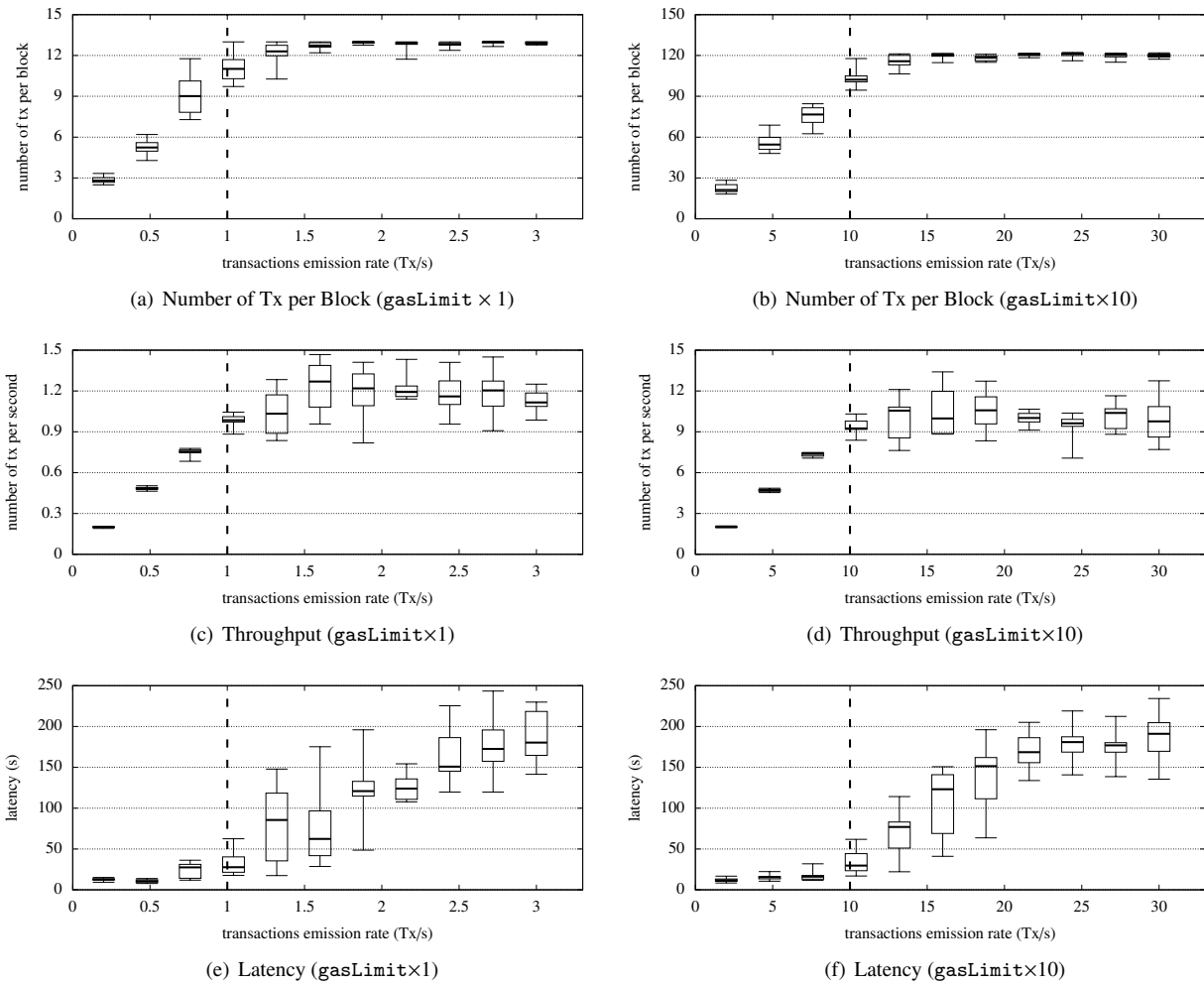


Figure 9: Experimental analysis of *capacity* of blockchain

tract is lighter to emit (the transaction fee is 894 159 gas instead of 1 148 305 gas as in the proposed FarMarket-related contracts), which should result in filling blocks with a greater number of contracts.

This is experimentally confirmed/validated in Figure 10 considering measurements of the *capacity* for different block sizes (defined in terms of `gasLimit`) and linear regressions between these values, stating the linearity assumed in the previous section. It can be observed that the slope rate is lower (by 19%) for the proposed approach compared with Tian (2017)'s smart contract, which is in line with the fact that the proposed contract is 28% heavier in gas transaction cost. This supports a link between the transaction cost of a contract and the slope (i.e., the number of transactions per unit of block size). Given this, a deeper analysis could provide

a prediction of the block size required to achieve a given throughput, for a given contract, such that the emission transaction rate remains below the blockchain *capacity*. In the implementation of Ethereum, the block size can be tuned by modifying the `gasLimit` of the blocks.

However, it is important to remember that increasing the block size can increase the propagation delays, and therefore, specific attention must be considered to ensure that the network capacity is sufficient. Furthermore, because the *capacity* linearly depends on the block generation throughput, it is also related to the delay in mining a block. By reconfiguring the feedback control of the difficulty (i.e., by modifying the blockchain implementation), such a delay can be adapted to support the expected *capacity*. Here, specific attention must be considered as it could promote the ap-

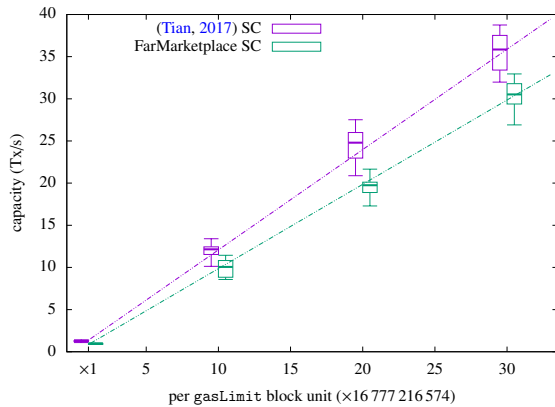


Figure 10: Comparison of *capacity* evolution for two smart contracts

pearance of ommers through the network. Finally, this *capacity* can be increased by optimizing the contract (i.e., by making it lighter). From a macroscopic perspective, the complexity of the algorithm writing data to the chain should be as low as possible (Wood, 2017). Using an optimizer such as GASOL (Albert et al., 2020) is an acceptable option to reduce contract gas fees in this respect.

4.4. Approach limitations

Even though the experiments were performed in a state near the steady regime for the mining delay, the network considered in this study was not subject to high network latency or data corruption. When using blockchain over the Internet, this delay could be more significant (e.g., approximately 12.6 seconds considering the Bitcoin chain (Decker and Wattenhofer, 2013)), which in certain applications could lead to end-to-end latency and throughput problems as discussed in (Fan et al., 2020; Bez et al., 2019). One of the major impacts of larger delays is the higher probability of the appearance of ommers related to the desynchronization effect. The feedback control on delay would respond by increasing the difficulty, and thus the delay of mining a block, which explains why the mining delay of the main Ethereum blockchain is approximately 14.4 seconds (Pierro, 2019).

We also stress the fact that in this study we did not consider the potential evolution of the `gasLimit`. Indeed, miners could be interested in increasing this limit (to decrease the number of blocks to mine) such that the *capacity* would evolve as defined previously. Clients such as Geth implement the limitation of the variation between two blocks as defined in (Wood, 2017) (approximately 0,1%). For 10 minute experiments (i.e.,

with an average of 52 blocks), it could correspond up to a 5% increase in the block size.

5. Conclusion, implications, and limitations

5.1. Conclusion

Research initiatives on how to integrate agriculture with blockchain technology remain in their infancy, with several outstanding research challenges and gaps (Hang et al., 2020; Zhao et al., 2019). Among these, as revealed in the literature review of the research presented in this paper, there is a requirement for blockchain-based farming marketplaces that support the trading of agricultural goods between farmers and interested third party stakeholders (e.g., food transformation companies, retailers), which should motivate a movement away from the centralized and monopolistic model that shapes today’s food value chain.

This study introduced such a blockchain-based farming marketplace, called “FarMarketplace”, a part of a larger ecosystem referred to as “FarMarket”. In this respect, trading (smart) contract templates between farmers, interested third-party consumers, and deliverers were specified. In addition to the specification of the FarMarket ecosystem, a comprehensive methodology was introduced to assist software solution integrators to better understand (and measure) what QoS performance a FarMarket-like ecosystem could achieve and support in the long run. A particular focus was given to the maximum achievable throughput (Tx/s) in the long run, which is referred to as *capacity* in this study. The experimental analyses presented in this paper should lead to interesting discussions regarding the critical aspects/interactions to be considered between blockchain- and infrastructure-related parameters.

5.2. Implications

This research presented three main theoretical implications. First, it contributes to the literature on smart farming (or e-agriculture) by proposing a thorough state-of-the-art approach for the use of blockchain technology, identifying the trends and gaps in the current research.

Secondly, it contributes to making agricultural and farming practices more sustainable in two respects: (i) it facilitates the emergence of local agriculture markets, thus encouraging agriculture and food sourcing and (ii) the nature of blockchain technology helps to prove that climate friendly requirements are met, as farmers are facing an increasing number of obligations for monitoring, verifying, and reporting according to sustainability requirements.

876 Thirdly, it contributes to the software development 923
877 community. To the best of our knowledge, there is 924
878 only limited research work that thoroughly discusses 925
879 the interactions between blockchain- and infrastructure- 926
880 related parameters, and how they influence the overall 927
881 (end-to-end) QoS performance. The experimental eval- 928
882 uation of the maximum achievable throughput (Tx/s) in 929
883 the long run (i.e., *capacity*) is a contribution of this re- 930
884 search work. 931

885 5.3. Limitations and Future research directions 933

886 Several limitations of our work should be addressed 934
887 and discussed. The first limitations, related to our 935
888 experiments, were identified and discussed in Section 936
889 4.4; therefore, we refer the reader to that section for 937
890 experiment-related limitations). 938

891 A second limitation relates to the proposed smart con- 939
892 tract templates, and particularly to the fact that the set 940
893 of data items considered in our templates could possi- 941
894 bly not cover all the trading requirements for the differ- 942
895 ent types of agricultural goods/markets to be sold/pur- 943
896 chased. Even though the *Description* parameter intro- 944
897 duced as part of our smart contracts is sufficiently 945
898 generic to be extended with any new information that 946
899 the farmer/seller could deem as relevant (only the hash 947
900 of the *Description* is added to the smart contract), it 948
901 would be convenient to adopt standardized metadata for 949
902 describing agricultural goods for enhanced interoper- 950
903 ability. Semantic- or ontology-based approaches could 951
904 be investigated and combined with blockchain-based 952
905 farming ecosystems (Bacco et al., 2019; Lokers et al., 953
906 2016).

907 Although beyond the scope of this research, one key 954
908 challenge lies in the adoption of blockchain-based so- 955
909 lutions by small and medium businesses. The reason 956
910 for this is twofold: (i) they frequently lack the expertise 957
911 to invest in blockchain (a common argument that can be 958
912 found in the literature is that there is no significant adop- 959
913 tion of blockchain technology outside of cryptocurrencies) 960
914 and (ii) clear feedback on the experience gained 961
915 from the deployment of blockchain-based systems is 962
916 limited owing to its recent emergence, although selected 963
917 reports have provided predictions on the potential gains; 964
918 see, e.g., IBM report¹¹ that states that blockchain can re- 965
919 duce the time required to trace the source of food from 966
920 seven days to 2.2 seconds. It is therefore imperative to 967
921 make blockchain infrastructures affordable and easy to 968
922 use in the near future.

¹¹<https://newsroom.ibm.com/How-Blockchain-Could-Mend-Our-Fractured-Global-Food-Supply-Chain>

Data privacy and security aspects related to 923
blockchain have not been discussed significantly in this 924
paper, although they are of importance in blockchain 925
applications. Indeed, by design, data inserted into a 926
blockchain cannot be erased. Furthermore, the strength 927
of a public blockchain is that everyone can download 928
and verify blocks and transactions, thus leaving room 929
for privacy concerns. Although sensitive data in the pro- 930
posed solution are stored apart from the blockchain (in 931
an external database called FarMarketDB), the “hash” 932
of that data is added to the blockchain (via smart con- 933
tract). Other more advanced solutions could be explored 934
in the future, similar to the ones proposed in (Kosba 935
et al, 2016; Bünz et al, 2019). 936

Our study of the parameters’ influence is also limited 937
by the chosen blockchain; Ethereum runs on the Geth 938
client. In fact, the diversity of consensus protocols and 939
chain parameters supporting existing blockchain tech- 940
nologies makes it difficult to objectively compare two 941
technologies. To do this effectively, our analysis should 942
be extended to other chains that function with other con- 943
sensus protocols. With this extension, a similar compar- 944
ison basis could be defined, enabling a better choice for 945
the blockchain infrastructure selection. 946

Finally, we must highlight the fact that the rela- 947
tion considered in Section 4.1 between “difficulty” and 948
“Blocktime” is based on an assumption (this assumption 949
being further detailed in Appendix A), and further re- 950
search should be performed to determine more accurate 951
numerical values (e.g., regarding the average difficulty 952
calculation). 953

954 Acknowledgments

955 This work was partly supported by the French PIA 956
project “Lorraine Université d’Excellence” reference 957
ANR-15-IDEX-04-LUE. We would like to express spe- 958
cial thanks to Mario Lezoche for his contribution 959
to the brainstorming challenges and ideas related to 960
blockchain-based farming.

961 Appendix A. Preliminary benchmark analysis

962 In this paper, certain relations were assumed, as in 963
Section 4.1. This appendix proposes an explanation 964
of these relations based on a probabilistic model of 965
Ethereum nodes. This model has been subjected to ex- 966
periments to verify the consistency of our benchmark 967
platform. Finally, a relation that establishes the time 968
and difficulty required to mine a block is discussed.

969 *Appendix A.1. Difficulty and Blocktime Relation*

In the context of Ethereum, the difficulty-related feedback control relies on an assumed relation given in (A.1), where τ is the average blocktime (i.e., the average delay to mine a block), d is the difficulty of the chain (relying on a statistical estimate of the number of hashes that must be generated to find a valid solution to mine a block), and $\#_{tot}$ is the network hashrate (i.e., the number of hashes realized by a node on a per second basis). As the blocktime is fixed to approximately ten seconds in Ethereum, this relation is frequently simplified by $d = 10 \times \#_{tot}$ (cf., Section 4.1).

$$\tau = \frac{d}{\#_{tot}} \quad (\text{A.1})$$

970 This relation can be interpreted as the mean of an exponential distribution of parameter $\lambda = \frac{\#_{tot}}{d}$. Indeed, as
 971 the mining of the block is comparable to a brute force 1005
 972 attack for a puzzle solution in a set sufficiently large, the 1006
 973 search of the solution can be modeled by a continuous 1007
 974 memoryless distribution. This model also provides an 1008
 975 explanation of the hashrate additivity property, as a network 1009
 976 composed of N nodes results in a system of $(n_i)_{i \in N}$ 1010
 977 independent nodes seeking the solution to the puzzle, 1011
 978 each having a given hashrate denoted by $\#_i$. Therefore, 1012
 979 for each node n_i , the random value denoted by X_i for 1013
 980 finding a solution follows an exponential parameter distribution $\frac{\#_i}{d}$. The network's random value X_{tot} to find a
 981 solution among all network nodes is therefore equal to
 982 $X_{tot} = \min(\{X_i\}_{i \in N})$ as a solution is found if and only
 983 if one node solves the puzzle, following an exponential
 984 network distribution $\frac{\sum_{i \in N} \#_i}{d}$. Overall, the hashrate of a
 985 network can be determined by summing the hashrate of
 986 the entire network.
 987
 988

989 *Appendix A.2. Validity of the model*

990 To verify the consistency of the proposed model and
 991 benchmark platform, the relation between blocktime
 992 and hashrate was tested. To achieve this, a blockchain
 993 with the same initial parameters (including the same
 994 difficulty) was performed with a variable number of
 995 hashrates. The experiment duration (ten minutes) was
 996 sufficiently short to neglect the change in difficulty due
 997 to the feedback control. In the experiment that produced
 998 the maximum number of blocks, which is more likely to
 999 be influenced by this control, 64 blocks were produced,
 1000 which could modify the difficulty in Ethereum accord-
 1001 ing to A.3 by up to 3% (this equation is further detailed
 1002 in the next section). Indeed, according to A.3, a block
 1003 can modify the difficulty of a block by a factor $\frac{1}{2048}$.
 1004 Thus, after 64 blocks, $\Delta d = \left| \left(1 \pm \frac{1}{2048}\right)^{64} - 1 \right| < 3.2\%$.

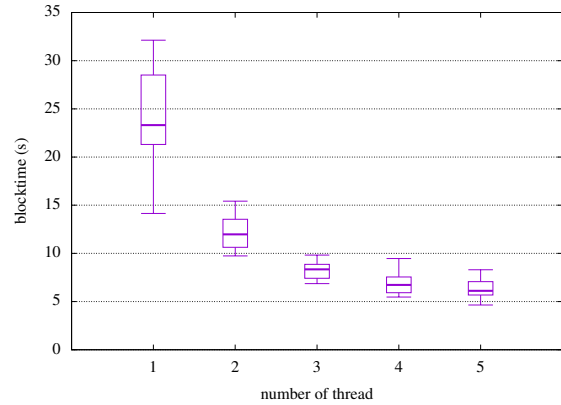


Figure A.11: Evolution of average blocktime according to number of threads

Furthermore, to eliminate the influence of the transfer time factor, the blockchain was executed on a single machine and the hashrate variation was performed by changing the number of threads allocated to the mining. Every thread used one core of the machine and proceeded independently via multi-threading. As long as the threads did not compete among themselves, they could be considered as having the same hashrate as they were executed on a similar core.

Given this hypothesis, the problem is equivalent to the relation given in A.2, where $\#_{thread}$ refers to the hashrate of a single thread that is assumed to be constant, and nb_{thread} to the number of threads used. Figure A.11 provides insight into the results obtained for an experiment realized ten times per number of threads. The relation between the blocktime and number of threads was inversely proportional. Figure A.12 stresses this finding by indicating a linear relation between the inverse of the delay and number of threads, following a linear regression with $\rho = 0.91$. When considering a large number of threads, this relation becomes obsolete owing to the fact that threads began to compete with each other on the same machine.

$$\tau = \frac{d}{\#_{thread} \times nb_{thread}} \quad (\text{A.2})$$

1014 *Appendix A.3. Difficulty Definition*

The definition of the difficulty of a block used in the current version of Ethereum has been extracted from yellow paper (Wood, 2017), and is given in (A.3) $\forall n >$

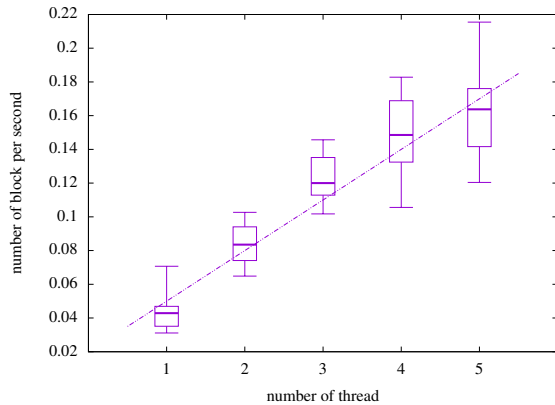


Figure A.12: Evolution of inverse of average blocktime according to number of threads

0:

$$H_d^n = \max\left(H_d^0, H_d^{n-1} + \left\lfloor \frac{H_d^{n-1}}{2048} \right\rfloor \times \max\left(y - \left\lfloor \frac{H_t^n - H_t^{n-1}}{9} \right\rfloor, -99\right) + \epsilon\right) \quad (\text{A.3})$$

$$y = \begin{cases} 1 & \text{if } H^{n-1} \text{ has no ommers} \\ 2 & \text{otherwise,} \end{cases}$$

where H^n refers to a block with a sequence number denoted by n , H_d^n is the difficulty of that block, H_t^n is the timestamp (in seconds) when the block was generated,

ϵ is the “difficulty bomb” designed to force users to update their chain (note that a high number of blocks – approximately 5 000 000 – must be considered to activate such a “bomb”), and y is a term depending on the appearance of ommers in the previous block. The maximum functions included in the formula ensure that the difficulty does not fall below the original value, while limiting its evolution speed. This relation can be interpreted as explained by (Wood, 2017): before the difficulty bomb, if no ommer appears, and if the delay between two blocks is between 9 and 18 seconds, the difficulty does not change. However, if the blocktime is not in this interval, the difficulty must be reduced or increased to favor the next blocktime in that interval.

References

Blandford, D., 2019. Supply Response to Increased Demand for Food. Global Challenges For Future Food And Agricultural Policies, 1, 61.

Zhao, G., Liu, S., Lopez, C., Lu, H., Elgueta, S., Chen, H., Boshkoska, B. M., 2019. Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions. Computers in Industry, 109, 83–99.

Rabah, K., 2018. Convergence of AI, IoT, big data and blockchain: a review. The Lake Institute Journal, 1(1), 1–18.

Lin, Y. P., Petway, J. R., Anthony, J., Mukhtar, H., Liao, S. W., Chou, C. F., Ho, Y. F., 2017. Blockchain: The evolutionary next step for ICT e-agriculture. Environments, 4(3), 50.

Aker, J. C., Ghosh, I., Burrell, J., 2016. The promise (and pitfalls) of ICT for agriculture initiatives. Agricultural Economics, 47(S1), 35–48.

Lezoche, M., Hernandez, J., Diaz, M. D. M. A., Panetto, H., Kacprzyk, J. (2020). Agri-food 4.0: a survey of the supply chains and technologies for the future agriculture. Computers in Industry, 116, 103187.

Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M. J., 2017. Big data in Smart Farming – a review. Agricultural Systems, 153, 69–80.

Vermesan, O., Friess, P., 2016. Digitising the industry-internet of things connecting the physical, digital and virtual worlds. River Publishers.

Krishnan, R. S., Julie, E. G., Robinson, Y. H., Raja, S., Kumar, R., Thong, P. H., 2020. Fuzzy Logic based Smart Irrigation System using Internet of Things. Journal of Cleaner Production, 252, 119902.

Kamble, S. S., Gunasekaran, A., Gawankar, S. A., 2020. Achieving sustainable performance in a data-driven agriculture supply chain: A review for research and applications. International Journal of Production Economics, 219, 179–194.

Klerkx, L., Jakku, E., Labarthe, P., 2019. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. NJAS-Wageningen Journal of Life Sciences, 100315.

Rose, D. C., Chilvers, J., 2018. Agriculture 4.0: Broadening responsible innovation in an era of smart farming. Frontiers in Sustainable Food Systems, 2, 87.

Pinto, M. M. A., Kovaleski, J. L., Yoshino, R. T., Pagani, R. N., 2019. Knowledge and technology transfer influencing the process of innovation in green supply chain management: A multicriteria model based on the DEMATEL Method. Sustainability, 11(12), 3485.

Tripoli, M., Schmidhuber, J., 2018. Emerging Opportunities for the Application of Blockchain in the Agri-food Industry. FAO and ICTSD: Rome and Geneva. Licence: CC BY-NC-SA, 3.

Bermeo-Almeida, O., Cardenas-Rodriguez, M., Samaniego-Cobo, T., Ferruzola-Gómez, E., Cabezas-Cabezas, R., Bazán-Vera, W., 2018. Blockchain in agriculture: A systematic literature review. In: International Conference on Technologies and Innovation, Guayaquil, Ecuador, November, 44–56.

Feng, H., Wang, X., Duan, Y., Zhang, J., Zhang, X., 2020. Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges. Journal of Cleaner Production, 121031.

Kamilaris, A., Fonts, A., Prenafeta-Boldú, F. X., 2019. The rise of blockchain technology in agriculture and food supply chains. Trends in Food Science & Technology, 91, 640–652.

Zheng, Z., Xie, S., Dai, H. N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4), 352–375.

Panarello, A., Tapas, N., Merlino, G., Longo, F., Puliafito, A., 2018. Blockchain and IoT integration: A systematic survey. Sensors, 18(8), 2575.

Nowiński, W., Kozma, M., 2017. How can blockchain technology disrupt the existing business models?. Entrepreneurial Business and Economics Review, 5(3), 173–188.

Al-Jaroodi, J., Mohamed, N., 2019. Blockchain in industries: A survey. IEEE Access, 7, 36500–36515.

Juma, H., Shaalan, K., Kamel, I., 2019. A Survey on Using Blockchain in Trade Supply Chain Solutions. IEEE Access, 7, 184115–184132.

- Xu, X., Weber, I., Staples, M., 2019. Case Study: AgriDigital. In: Architecture for Blockchain Applications. Springer, Cham, 239–255.
- Dujak, D., Sajter, D., 2019. Blockchain applications in supply chain. In: SMART supply network. Springer, Cham, 21–46.
- Hang, L., Ullah, I., Kim, D. H., 2020. A secure fish farm platform based on blockchain for agriculture data integrity. *Computers and Electronics in Agriculture*, 170, 105251.
- Galvez, J. F., Mejuto, J. C., Simal-Gandara, J., 2018. Future challenges on the use of blockchain for food traceability analysis. *TrAC Trends in Analytical Chemistry*, 107, 222–232.
- Thomason, J., Ahmad, M., Bronder, P., Hoyt, E., Pocock, S., Bouteloupe, J., Donaghy, K., Huysman, D., Willenberg, T., Joakim, B., 2018. Blockchain – Powering and empowering the poor in developing countries. In: Transforming climate finance and green investment with blockchains. Academic Press, 137–152.
- Bumblauskas, D., Mann, A., Dugan, B., Rittmer, J. (2020). A blockchain use case in food distribution: Do you know where your food has been?. *International Journal of Information Management*, 52, 102008.
- Pinna, A., Ibba, S., 2018. A blockchain-based Decentralized System for proper handling of temporary Employment contracts. In: Science and information conference. Springer, Cham, 1231–1243.
- Devi, M. S., Suguna, R., Joshi, A. S., Bagate, R. A., 2019. Design of IoT Blockchain Based Smart Agriculture for Enlightening Safety and Security. In: International Conference on Emerging Technologies in Computer Engineering. Springer, Jaipur, India, February, 7–19.
- Patil, A. S., Tama, B. A., Park, Y., Rhee, K. H., 2017. A framework for blockchain based secure smart green house farming. In: Advances in Computer Science and Ubiquitous Computing. Springer, 1162–1167.
- Tse, D., Zhang, B., Yang, Y., Cheng, C., Mu, H., 2017. Blockchain application in food supply information security. In: IEEE International Conference on Industrial Engineering and Engineering Management. Singapore, Singapore, December, 1357–1361.
- Davcev, D., Kocarev, L., Carbone, A., Stankovski, V., Mitreski, K., 2018. Blockchain-based Distributed Cloud/Fog Platform for IoT Supply Chain Management. In 8th international conference on advances in computing, electronics and electrical technology. Kuala Lumpur, Malaysia, February, 51–58.
- Lucena, P., Binotto, A. P., Momo, F. D. S., Kim, H., 2018. A case study for grain quality assurance tracking based on a Blockchain business network. arXiv preprint arXiv:1803.07877.
- Mao, D., Hao, Z., Wang, F., Li, H., 2019. Novel automatic food trading system using consortium blockchain. *Arabian Journal for Science and Engineering*, 44(4), 3439–3455.
- Tian, F., 2017. A supply chain traceability system for food safety based on HACCP, Blockchain & Internet of Things. In: International conference on service systems and service management. Dalian, China, June, 1–6.
- Bore, N., Kinai, A., Waweru, P., Wambugu, I., Mutahi, J., Kemunto, E., Weldemariam, K., 2020. ADW: Blockchain-enabled Small-scale Farm Digitization. arXiv preprint arXiv:2003.06862.
- Stefanova, M., Salampasis, M., 2019. Farm-to-fork traceability: Blockchain meets agri-food supply chain. In: 12th European Federation for Information Technology in Agriculture, Food and the Environment. Rhodes, Greece, June.
- Leng, K., Bi, Y., Jing, L., Fu, H. C., Van Nieuwenhuyse, I., 2018. Research on agricultural supply chain system with double chain architecture based on blockchain technology. *Future Generation Computer Systems*, 86, 641–649.
- Kumar, M. V., Iyengar, N. C. S., 2017. A framework for Blockchain technology in rice supply chain management. *Advanced science and technology letters*, 146, 125–130.
- Iqbal, R., Butt, T. A., 2020. Safe farming as a service of blockchain-based supply chain management for improved transparency. *Cluster Computing*, 1–12.
- George, R. V., Harsh, H. O., Ray, P., Babu, A. K., 2019. Food quality traceability prototype for restaurants using blockchain and food quality data index. *Journal of Cleaner Production*, 240, 118021.
- Caro, M. P., Ali, M. S., Vecchio, M., Giuffreda, R., 2018. Blockchain-based traceability in Agri-Food supply chain management: A practical implementation. In 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany. Borgo San Luigi, Italy, May, 1–4.
- Surasak, T., Wattanavichean, N., Preuksakarn, C., Huang, S. C. H., 2019. Thai Agriculture Products Traceability System using Blockchain and Internet of Things. Editorial Preface From the Desk of Managing Editor, 10(9), 15.
- Malik, S., Kanhere, S. S., Jurdak, R., 2018. Productchain: Scalable blockchain framework to support provenance in supply chains. In: 17th IEEE International Symposium on Network Computing and Applications. Cambridge, USA, October, 1–10.
- Hua, J., Wang, X., Kang, M., Wang, H., Wang, F. Y., 2018. Blockchain based provenance for agricultural products: A distributed platform with duplicated and shared bookkeeping. In: IEEE Intelligent Vehicles Symposium IV. Changshu, Suzhou, China, June, 97–101.
- Shakhbulatov, D., Arora, A., Dong, Z., Rojas-Cessa, R., 2019. Blockchain Implementation for Analysis of Carbon Footprint across Food Supply Chain. In: IEEE International Conference on Blockchain. Atlanta, USA, July, 546–551.
- Lin, J., Shen, Z., Zhang, A., Chai, Y., 2018. Blockchain and IoT based food traceability for smart agriculture. In: 3rd International Conference on Crowd Science and Engineering. Vancouver, Canada, July, 1–6.
- Reddy, G. B., Kumar, K. R., 2020. Quality Improvement in Organic Food Supply Chain Using Blockchain Technology. In: Innovative Product Design and Intelligent Manufacturing Systems. Springer, Singapore, 887–896.
- Xie, C., Sun, Y., Luo, H., 2017. Secured data storage scheme based on blockchain for agricultural products tracking. In: 3rd International Conference on Big Data Computing and Communications. Yonago, Japan, July, 45–50.
- Papa, S. F., 2017. Use of Blockchain technology in agribusiness: transparency and monitoring in agricultural trade. In: International Conference on Management Science and Management Innovation. Suzhou, China, June.
- Awan, S. H., Ahmed, S., Safwan, N., Najam, Z., Hashim, M. Z., Safdar, T. Role of Internet of Things (IoT) with Blockchain Technology for the Development of Smart Farming. *Journal of Mechanics of Continua and Mathematical Sciences*, 14, 170–188.
- Laux, C. M., Hurburgh Jr, C. R., 2012. Using quality management systems for food traceability. *Journal of Industrial Technology*, 26(3), 1.
- Udokwu, C., Kormiltsyn, A., Thangalimodzi, K., Norta, A., 2018. The state of the art for blockchain-enabled smart-contract applications in the organization. In: Ivannikov Ispras Open Conference. Moscow, Russia, November, 137–144.
- Xie, J., Tang, H., Huang, T., Yu, F. R., Xie, R., Liu, J., Liu, Y., 2019. A survey of blockchain technology applied to smart cities: Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(3), 2794–2830.
- Sajana, P., Sindhu, M., Sethumadhavan, M., 2018. On blockchain applications: Hyperledger fabric and Ethereum. *International Journal of Pure and Applied Mathematics*, 118(18), 2965–2970.
- Ali, M. S., Vecchio, M., Pincheira, M., Dolui, K., Antonelli, F., Rehmani, M. H., 2018. Applications of blockchains in the Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 21(2), 1676–1717.

1236 Xiao, Y., Zhang, N., Lou, W., Hou, Y. T., 2020. A survey of distributed
1237 consensus protocols for blockchain networks. *IEEE Communica-*
1238 *tions Surveys & Tutorials*. DOI: 10.1109/COMST.2020.2969706

1239 Vukolić, M., 2015. The quest for scalable blockchain fabric: Proof-
1240 of-work vs. BFT replication. In: *International workshop on open*
1241 *problems in network security*. Zurich, Switzerland, October, 112–
1242 125.

1243 Pierro, G. A., Rocha, H., 2019. The influence factors on ethereum
1244 transaction fees. In: *2nd IEEE/ACM International Workshop on*
1245 *Emerging Trends in Software Engineering for Blockchain*. Mon-
1246 treal, Canada, May, 24–31.

1247 Wood, G., 2014. Ethereum: A secure decentralised generalised trans-
1248 action ledger. *Ethereum project yellow paper*, 151(2014), 1–32.

1249 Albert, E., Correas, J., Gordillo, P., Román-Díez, G., Rubio, A., 2020.
1250 GASOL: Gas analysis and optimization for Ethereum smart con-
1251 tracts. In: *International Conference on Tools and Algorithms for*
1252 *the Construction and Analysis of Systems*. Dublin, Ireland, 118–
1253 125.

1254 Decker, C., Wattenhofer, R., 2013. Information propagation in the bit-
1255 coin network. In: *IEEE International Conference on Peer-to-Peer*
1256 *Computing*. Trento, Italy, September, 1–10.

1257 Bacco, M., Barsocchi, P., Ferro, E., Gotta, A., Ruggeri, M., 2019.
1258 The Digitisation of Agriculture: a Survey of Research Activities
1259 on Smart Farming. *Array*, 3, 100009.

1260 Lokers, R., Knapen, R., Janssen, S., van Randen, Y., Jansen, J., 2016.
1261 Analysis of Big Data technologies for use in agro-environmental
1262 science. *Environmental Modelling & Software*, 84, 494–504.

1263 Fan, C., Ghaemi, S., Khazaei, H. & Musilek, P, 2020. Performance
1264 Evaluation of Blockchain Systems: A Systematic Survey. *Ac-*
1265 *cess*,8, 126927–126950

1266 Bez, M., Fornari, G. & Vardanega, T. The scalability challenge of
1267 ethereum: An initial quantitative analysis.

1268 Kosba, A., Miller, A., Shi, E., Wen, Z. & Papamanthou, C. Hawk: The
1269 Blockchain Model of Cryptography and Privacy-Preserving Smart
1270 Contracts.

1271 Büinz, B., Agrawal, S., Zamani, M. & Boneh, D. Zether: Towards
1272 Privacy in a Smart Contract World.

1273 Sumathi, M. & Sangeetha, S.: Blockchain Based Sensitive Attribute
1274 Storage and Access Monitoring in Banking System.

1275 Singh, Nikita & Vardhan, Manu: Distributed Ledger Technology
1276 based Property Transaction System with Support for IoT Devices.

1277 Ramli, Carroline Dewi Puspa Kencana & Nielson, Hanne Riis & Niel-
1278 son, Flemming: The logic of XACML. *Science of Computer Pro-*
1279 *gramming*, 83.

1280 Di Francesco Maesa, Damiano & Mori, Paolo & Ricci, Laura:
1281 Blockchain Based Access Control.

1282 Esposito, Christian & Ficco, Massimo & Gupta, B B: Blockchain-
1283 based authentication and authorization for smart city applications.