Journal of Cleaner Production 306 (2021) 127055

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Innovative blockchain-based farming marketplace and smart contract performance evaluation

Guilain Leduc *, Sylvain Kubler, Jean-Philippe Georges

Université de Lorraine, CNRS, CRAN, F-54000, Nancy, France

ARTICLE INFO

Article history:
Received 29 April 2020
Received in revised form
1 April 2021
Accepted 6 April 2021
Available online 16 April 2021

Handling editor: Cecilia Maria Villas Bôas de Almeida

Keywords:
Blockchain
Trading
Smart agriculture
Network performance
Quality of service

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.

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/precision[1] (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., 2020; 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 have appeared around the world (Lin et al., 2017; Kamilaris et al., 2019), wherein different research questions at the hardware, software, network, and governance levels have been addressed.

Whereas the majority of the research has focused on investigating blockchain-based solutions for the enhanced tracking and traceability of agricultural goods, as is further analyzed and

* Corresponding author.
E-mail addresses: guilain.leduc@univ-lorraine.fr (G. Leduc), s.kubler@univ-lorraine.fr (S. Kubler), jean-philippe.georges@univ-lorraine.fr (J.-P. Georges).

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

https://doi.org/10.1016/j.jclepro.2021.127055
0959-6526/

discussed in Section 2, limited research has been undertaken on the design of innovative smart farming digital marketplaces to support the trading of agricultural goods between farmers and interested third party stakeholders (e.g., food transformation companies, retailers, and other farmers). To overcome this gap in research, this paper presents a novel digital marketplace called "FarMarketplace". FarMarketplace fully exploits the advantages of blockchain capabilities by proposing generic, yet detailed representations of trading (smart) contract templates between farmers, interested third-party consumers, and deliverers. Compared with the current literature, FarMarketplace is innovative in three respects.

- The evaluation of this blockchain solution facilitates a methodology to benchmark a blockchain system. It focuses most notably on the allowable capacity offered by the blockchain immediately before saturation. Hence, this evaluation is based on the expected contract emission throughput and its latency according to the block size. Consequently, the notion of *capacity*, the maximum throughout that the chain can support is introduced. Phenomena around the *capacity* are also presented, and the methodology is applied for a specific contract.
- Only limited blockchain-based farming frameworks/ecosystems focus on trading and propose any kind of digital marketplaces where farmers/industries, deliverers, and retailers can discover each other and trade agricultural goods and delivery services;
- The majority of the studies do not provide sufficient details regarding the performance of their proposed system (more than 50% do not evaluate any metrics), and to the best of our knowledge, no study has ever defined a comprehensive methodology to assist software solution integrators understand 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 background is given in Section 2.1. In Section 2.2, past and ongoing blockchain-based farming/agricultural initiatives are reviewed and analyzed. Based on this literature 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 respect, a number of consortia are working on the design of decentralized digital marketplaces in different sectors such as healthcare, logistics, energy, construction, agriculture, and telecommunication (Al-Jaroodi and Mohamed, 2019). Domain-independent initiatives are also being identified, such as Trusted IoT Alliance[2] and IOTA Foundation,[3] Enterprise Ethereum Alliance (EEA), and Flowchain.[4] All these initiatives promote 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 Fig. 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 Skuchain,[5] Provenance,[6] AgriDigital,[7] (Xu et al., 2019), and Farm Share[8] 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). Fig. 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.

[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.

[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.

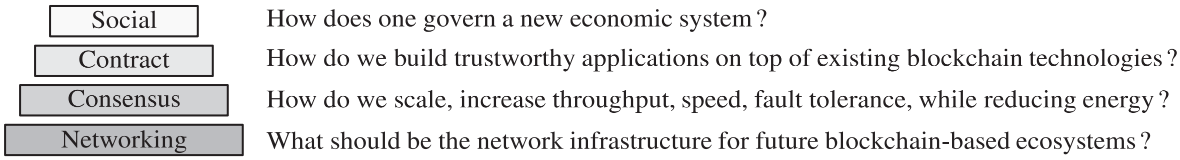


Fig. 1. Blockchain Architecture Stack and associated research questions.

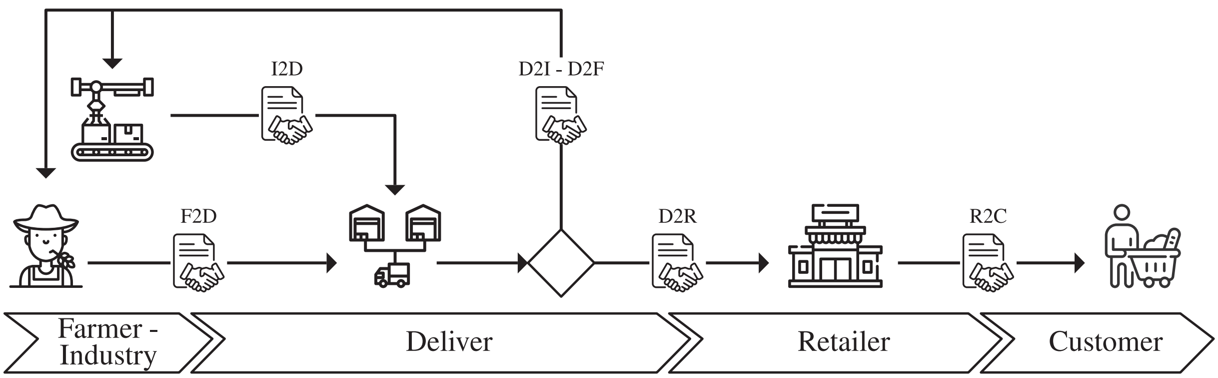


Fig. 2. Traditional food chain.

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

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	☐	☐	☐	☐	Ethereum	–
Devi et al. (2019)	Track	N/F	☐	☐	☐	☐	Ethereum	1. Latency
Patil et al. (2017)	Track	–	☒	☐	☐	☐	N/S	–
Tse et al. (2017)	Trace	–	☒	☐	☐	☐	N/S	–
Davcev et al. (2018)	Track	N/F	☒	☐	☐	☐	Hyperledger	–
Hang et al. (2020)	Track	F	☒	☐	☐	☐	Hyperledger (v1.4.3)	1. Throughput
Lin et al. (2017)	Trace & Track	N/F	☒	☐	☐	☐	–	–
Lucena et al. (2018)	Track	N/F	☒	☒	☐	☐	Hyperledger	–
Mao et al. (2019)	Trade	N/F	☒	☒	☒	☐	Own (FTSCON)	1. Exec. time 2. Merchant profit, 3. Security
Tian (2017)	Trace	N/F	☒	☒	☒	☐	Ethereum	–
Bore et al. (2020)	Trade	N/F	☒	☒	☒	☐	Hyperledger	1. Throughput 2. Latency
Stefanova and Salampasis (2019)	Trace	N/F	☒	☒	☒	☐	Hyperledger	–
Leng et al. (2018)	Trade	N/F	☒	☒	☒	☐	N/S	1. Throughput 2. Latency
Kumar and Iyengar (2017)	Trace & Track	N/F	☒	☒	☒	☐	–	–
Iqbal and Butt (2020)	Track	N/F	☒	☒	☒	☐	–	1. ZigBee-related
George et al. (2019)	Trace & Track	N/F	☒	☒	☒	☒	–	–
Caro et al. (2018)	Trace & Track	N/F	☒	☒	☒	☒	Ethereum & Hyperledger	1. Throughput 2. Latency 3. CPU
Surasak et al. (2019)	Trace & Track	N/F	☒	☒	☒	☒	SQL-based	–
Bumblauskas et al. (2020)	Trace & Track	N/F	☒	☒	☒	☒	Hyperledger	–
Malik et al. (2018)	Trace	F	☒	☒	☒	☒	Hyperledger	1. Time
Hua et al. (2018)	Trace & Track	F	☒	☒	☒	☒	N/S	–
Shakhbulatov et al. (2019)	Track	F	☒	☒	☒	☒	Raft-like consensus	1. Throughput 2. Time
Lin et al. (2018)	Trace	N/F	☒	☒	☒	☒	–	–
Reddy and Kumar (2020)	Trace & Track	N/F	☒	☒	☒	☒	–	–
Xie et al. (2017)	Track	F	☒	☒	☒	☒	Ethereum (v1.9)	1. Throughput
Papa (2017)	Trace	N/F	☒	☒	☒	☒	–	–
Awan et al. (2019)	Trace & Track	N/F	☒	☒	☒	☒	N/S	1. Throughput

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.

First, it can be observed that all the reported studies have been published in the last three years, which confirms the growing attention paid to blockchain in the agricultural sector. Moreover, the majority of the studies (23 out of the 27 reported in Table 1) employ blockchain for food traceability and/or tracking purposes. The four other studies use blockchain to automate temporary employment contracts between the farmers and labor contractors (Pinna and Ibba, 2018) and allow for agricultural resource trading between farmers, deliverers, and retailers (Leng et al., 2018; Mao et al., 2019; Bore et al., 2020).

Secondly, virtually all the reported studies exploit the smart contract capabilities to achieve the above-mentioned objectives (i.e., to meet traceability, tracking, and trading requirements); 17 of the 27 studies focus on − *or fulfill to be more precise* − market interactions between farmers/industries, deliverers, and retailers (i.e., F−I2D, D2F−I, R2C). Of these 17 studies, 12 extend the traceability, tracking, or trading facilities to the entire food lifecycle (i.e., covering R2C interactions). It should be noted that the reported studies do not necessarily track/trace the same food system features. Indeed, certain studies such as (Hang et al., 2020; Bumblauskas et al., 2020; Devi et al., 2019; Surasak et al., 2019; Lin et al., 2018) track the environmental background information of a food item using sensor-like devices (e.g., amount of pesticides used, temperature evolution), whereas other studies track other supply chain information such as (i) incident details throughout the crop harvesting process (Iqbal and Butt, 2020), (ii) carbon footprint at food production and transportation levels (Shakhbulatov et al., 2019), and (iii) food quality evolution (Davcev et al., 2018; George et al., 2019).

Thirdly, reviewing what blockchain technologies have been considered in the reported studies (see column "Platform" in Table 1), Ethereum and Hyperledger Fabric are the most widely adopted solutions (the former being used in five studies, the latter in five). This is not surprising as they are both market share leaders (50% of the implemented projects being hosted on these platforms) (Udokwu et al., 2018). However, interestingly, one could question why studies aiming to achieve similar goals opt for one or the other? Indeed, whereas Ethereum is more suitable for permissionless distributed ledgers, Hyperledger is more suited to permissioned blockchains (Xie et al., 2019; Sajana et al., 2018). To answer this question, a more in-depth analysis of these studies should be performed to identify the exact system requirements and constraints.

Finally, it can be observed in Table 1 (see column "Performance") that less than half of the reviewed studies performed experimental evaluations of their solutions. In our opinion, this clearly indicates that blockchain-based farming remains in its infancy, where the focus is more on architectural and functional design choices than on performance benchmarking. For studies evaluating the performance of their solution, throughput and latency are the most used metrics, considered in 65% and 50%, respectively, of the reviewed literature). Throughput corresponds to the number of successful transactions per second (a transaction being successful if it has been validated and committed to a new block); latency corresponds to the delay between the emission of a transaction and its commitment to a new block. It can be observed that only a small number of studies evaluated security aspects. The main reason for this is that the majority of the proposed solutions rely on off-the-shelf blockchain solutions, whose security performance − *which is characterized by the number of trusted participants required to secure the blockchain* − has been widely studied and described in the literature (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.
3. The vast majority of the studies do not provide sufficient details regarding the performance of the proposed system (more than 50% of the studies do not evaluate any metrics). This lack of comprehensive evaluation in the studies, combined with the lack of details regarding the implemented smart contracts (only 20% of the studies provide relevant details) and lack of details regarding the blockchain configuration, which has a direct impact on the overall system performance, makes it difficult to compare existing blockchain-based farming frameworks.

Given the above findings, this paper advances the current state-of-the-art in two respects. First, a novel digital marketplace for agricultural product trading purposes is proposed, which is in agreement with research by Mao et al. (2019); Bore et al. (2020);

Leng et al. (2018). Secondly, a representation of the interactions that occur between blockchain- and/or infrastructure-related parameters is presented. This representation not only provides software solution integrators with a holistic overview of possible interactions, but also facilitates the analysis of the system (QoS) performance limitations in the long run.

3. FarMarket ecosystem

The digital FarMarletplace 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 Fig. 3. This ecosystem is designed to:

- collect agricultural bids/contracts published by farmers or other like-minded providers (see in Fig. 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 deliver the asset associated to the bid/contract (see ④ and ⑤), upon which they will be paid;
- allow all stakeholders to evaluate the service quality, namely (i) the consumer can evaluate the quality of the delivery service (e.g., punctuality, professionalism) and the received agricultural goods, (ii) the deliverer can evaluate the quality of the farmer and consumer (e.g., punctuality, accuracy of the specified location); and (iii) the farmer can evaluate the quality of the delivery service.

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

1. *FarMarketchain*: This refers to the blockchain and associated smart contracts. A database, denoted by DB in Fig. 3, functions with blockchain to avoid storing long chains of characters in the blockchain itself, which is costly (only the hash of the corresponding chain is added to the blockchain).
2. *FarMarketplace*: This refers to the digital marketplace platform. It hosts the blockchain and DB, and has the role of intermediary between the different ecosystem stakeholders.
3. *FarMarketApp*: This refers to the App that allows stakeholders to benefit from the set of services offered by the FarMarket ecosystem.

Using an exterior database to store data is a common practice in blockchain development. Indeed, it may become expensive to store raw information in a distributed ledger, as each transaction usually implies a fee, and storing only the hash of information stored in a database (allowing for verifying the data integrity by comparing that hash at any given time) is a widely adopted alternative. This database is, in our case, a server but can be substituted by a private cloud Sumathi and Sangeetha (2020) or IPFS (InterPlanetary File System) to allow for fully decentralized peer-to-peer framework Singh and Vardhan (2019). As the access control of data is not the main focus of this paper, all data is freely accessible, although more advanced access control strategies could be adopted in the future, as defining a XACML politic Ramli et al. (2014) or adopting a blockchain-based solution Di Francesco Maesa et al. (2017); Esposito et al. (2021).

3.2. Smart farming contracts

The set of interactions (or communications) occurring between the previously introduced building blocks and stakeholders are

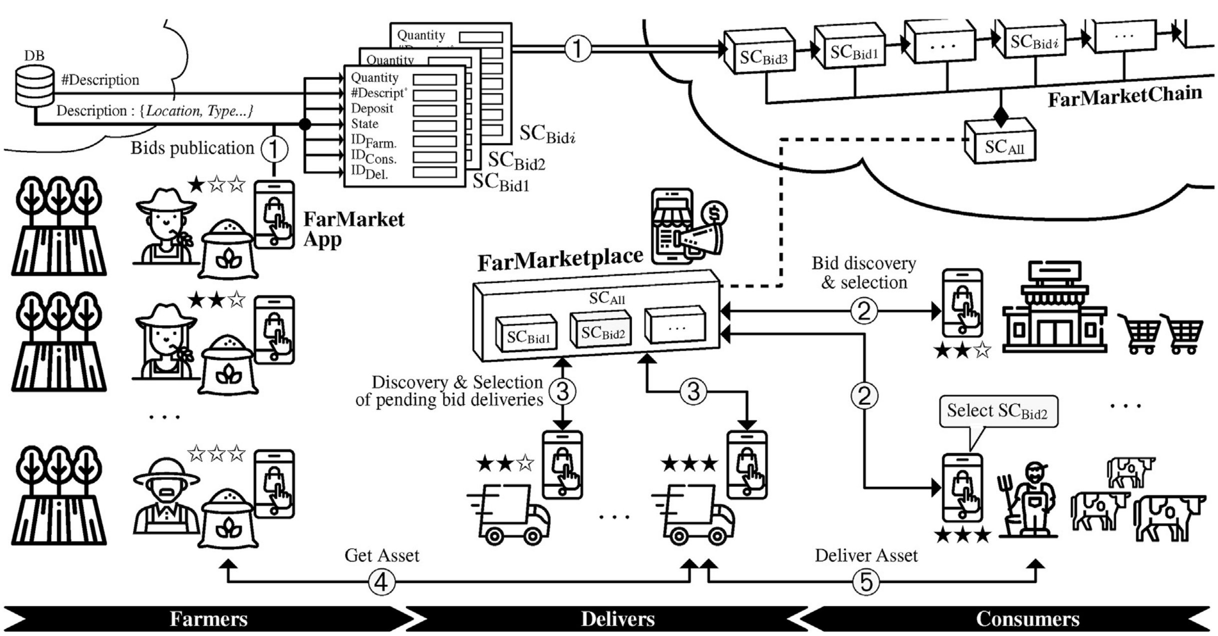


Fig. 3. Overview of the FarMarket ecosystem and associated interactions between stakeholders and supporting marketplace.

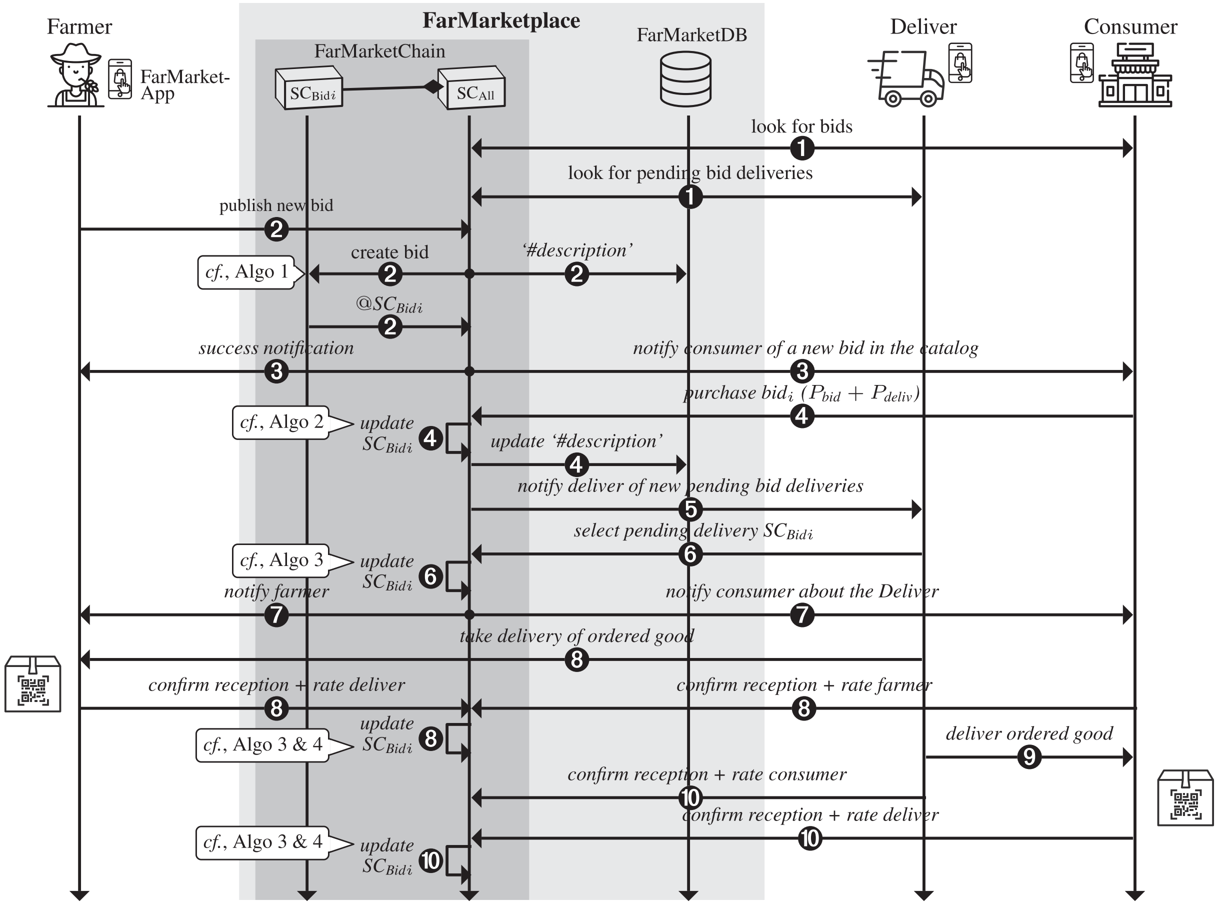


Fig. 4. Messaging protocol supporting FarMarket ecosystem.

further detailed in Fig. 4 in the form of a sequence diagram.

First, consumers and deliverers can look for bids and pending delivery offers (*cf.*, ❶ in Fig. 4), and possibly subscribe to the FarMarketplace platform to be notified 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 specifying – via *the FarMarketApp* – information related to their bid (*cf.*, ❷). This action calls the bid creation function 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 *FarMarketApp* 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 parameters, 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

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 *priceBid()* 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 Figs. 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 *User 1* is regarding the 'service' delivered by (or the behavior of) *User 2*. All possible rating score combinations are contained in a set denoted by $\mathscr{R} = \{R_{Far\rightarrow Del}, R_{Del\rightarrow Far}, R_{Del\rightarrow Con}, R_{Con\rightarrow Far}, R_{Con\rightarrow Del}\}$
State Variables	SC_{Bidi}^{state}	State variable referring to the state of contract SC_{Bidi} at a given point in time. Possible states are {*Available, WaitForDeliverer, WaitForDelivery, OnDelivery, Delivered*}
	$SCBidi^{ID_{User}}$	State variable referring to a given stakeholder (*cf.*, ID_{User})
	$SCBidi^{\#Desc}$	State variable referring to the hash of the bid's description (*cf.*, "Desc"), obtained using the *priceBid(...)* function
	$SC_{Bidi}^{balance}$	State variable referring to the balance of the smart contract/bid i
	$SC_{Bidi}^{P_{Far}}$, $SC_{Bidi}^{P_{Del}}$	State variables referring to prices to be paid to the farmer and deliverer (*cf.*, P_{Far}, P_{Del})
	$SCBidi^{R_{User1\rightarrow User2}}$	State variable referring to the satisfaction rating scores previously described (*cf.*, $\mathscr{R}$)

balance, and (v) contract state. The input parameters communicated by *FarMarketApp* and derived state variables are summarized in Table 2.

Algorithm 1. create_SC_{Bidi}

Algorithm 1: create_SC_{Bidi}

Input : ID_{User}, Desc
1 $SC_{Bidi}^{ID_{Far}} \leftarrow ID_{User}$; // Initialize contract's owner ID
2 $SC_{Bidi}^{\#Desc} \leftarrow hash(\mathrm{Desc})$; // Compute #Desc value
3 $SC_{Bidi}^{P_{Far}} \leftarrow priceBid(\mathrm{Desc}_{quant}, \mathrm{Desc}_{type} \ldots)$; // Compute bid price
4 $SC_{Bidi}^{balance} \leftarrow 0$; // Initialize bid's balance
5 $SC_{Bidi}^{state} \leftarrow$ Available; // Initialize contract state
Output: SC_{Bidi}

Algorithm 2. purchase_SC_{Bidi}

Algorithm 2: purchase_SC_{Bidi}

Input : $loc^{ID_{Con}}$, ID_{User}, Desc, amount, P_{Del}
1 **if** $SC_{Bidi}^{state} == Available \;\&\; amount == \left(SC_{Bidi}^{P_{Far}} + P_{Del}\right)$ **then**
2 $SC_{Bidi}^{ID_{Con}} \leftarrow ID_{User}$; // Update consumer's location
3 $SC_{Bidi}^{P_{Del}} \leftarrow P_{Del}$; // Set service delivery price
4 $\mathrm{Desc} \leftarrow \mathrm{Desc} \cup \{loc^{ID_{Con}}\}$; // Update DB description
5 $SC_{Bidi}^{\#Desc} \leftarrow hash(\mathrm{Desc})$; // Compute new #Desc value
6 $SC_{Bidi}^{balance} \leftarrow$ amount; // Deposit
7 $ID_{Con}^{balance} \leftarrow ID_{Con}^{balance} -$ amount; // Update balance
8 $SC_{Bidi}^{state} \leftarrow$ WaitForDeliverer; // Update SC state
Output: SC_{Bidi}

At this stage, the contract is available on the marketplace and consumers have been notified of its existence (❸). When a consumer selects a bid for purchase (cf., ❹), the purchase function of SC_{Bidi} is executed, as detailed in Algorithm 2. The purpose of this function/algorithm is to verify that the contract is in the correct state (should be Available for purchase) and that the consumer has sufficient money in her/his digital wallet. The amount of money required to purchase the contract should be equal to the bid price plus the service delivery price, which is denoted by P_{Del}. Note that P_{Del} is an input of the purchase function, meaning that it is computed outside the smart contract[9] and then added to the corresponding state variable (cf., line 3 of Algorithm 2). If these conditions are satisfied, the following state variables are updated: (i) contract's subscriber/beneficiary with the consumer's ID; (ii) contract delivery price; (iii) consumer's location, which is part of the bid description; (iv) contract balance credited with the required amount; and (v) contract state set to WaitForDeliverer. In addition, the balance of the consumer's wallet is updated accordingly (cf., line 7 in Algorithm 2). Once the contract state has been updated to *WaitForDeliverer*, deliverers who have subscribed to pending bid delivery offers are notified (cf., ❺).

Algorithm 3. delivery_SC_{Bidi}

Algorithm 3: delivery_SC_{Bidi}

Input : ID_{User}
1 **if** $SC_{Bidi}^{state} == WaitForDeliverer$ **then**
2 $SC_{Bidi}^{ID_{Del}} \leftarrow ID_{User}$; // Set deliverer's contract
3 $SC_{Bidi}^{state} \leftarrow$ WaitForDelivery ; // Update SC state
4 **else if** $SC_{Bidi}^{state} == WaitForDelivery \;\&\; SC_{Bidi}^{ID_{Del}} == ID_{\mathrm{User}}$ **then**
5 $SC_{Bidi}^{state} \leftarrow$ OnDelivering ; // Update SC state
6 **else if** $SC_{Bidi}^{state} == OnDelivering \;\&\; SC_{Bidi}^{ID_{Con}} == ID_{\mathrm{User}}$ **then**
7 $SC_{Bidi}^{state} \leftarrow$ Delivered; // Update SC state
8 $ID_{Farm}^{balance} \leftarrow ID_{Farm}^{balance} + SC_{Bidi}^{P_{bid}}$; // Update farmer balance
9 $ID_{Del}^{balance} \leftarrow ID_{Del}^{balance} + SC_{Bidi}^{P_{Del}}$; // Update deliverer balance
10 $SC_{Bidi}^{balance} \leftarrow 0$; // Set SC balance to zero
Output: SC_{Bidi}

When a deliverer selects a pending bid delivery offer (cf., ❻), the delivery function of SC_{Bidi} is executed, as detailed in Algorithm 3. This function/algorithm verifies the contract state. If state is *WaitForDeliverer*, then the deliverer becomes the official delivery service provider (cf., line 2 in Algorithm 3), the contract state becomes *WaitForDelivery*, and both the consumer and farmer are notified of the deliverer's identity (cf., ❼). At this stage, the deliverer must take delivery of the ordered goods (cf., ❽). At the moment of exchanging the goods, the deliverer and farmer must both confirm the successful reception, which in practice, results in the call of the delivery_SC_{Bidi} function, leading to a change in the contract state to *OnDelivering* (cf., lines 4–5 in Algorithm 3). In the final stage, the deliverer delivers the goods to the consumer (cf., ❾). Both confirm the successful delivery/reception (cf., ❿), which leads to a change in the contract state to Delivered (cf., lines 6–7 in Algorithm 3), following which payments are made to the farmer and deliverer, and the balances updated accordingly (cf., lines 8–10).

Algorithm 4. rating_SC_{Bidi}

Algorithm 4: rating_SC_{Bidi}

Input : $ID_{\mathrm{User}}, \mathcal{R}$; // Consumer location
1 **if** $SC_{Bidi}^{state} == Delivered \;\&\; ID_{User} == SC_{Bidi}^{ID_{Far}}$ **then**
2 $SC_{Bidi}^{\mathcal{R}_{Far \to Del}} \leftarrow \mathcal{R}_{Far \to Del}$; // Set rating score
3 **else if** $SC_{Bidi}^{state} == Delivered \;\&\; ID_{User} == SC_{Bidi}^{ID_{Del}}$ **then**
4 $SC_{Bidi}^{\mathcal{R}_{Del \to Far}} \leftarrow \mathcal{R}_{Del \to Far}$; // Set rating score
5 $SC_{Bidi}^{\mathcal{R}_{Del \to Con}} \leftarrow \mathcal{R}_{Del \to Con}$; // Set rating score
6 **else if** $SC_{Bidi}^{state} == Delivered \;\&\; ID_{User} == SC_{Bidi}^{ID_{Con}}$ **then**
7 $SC_{Bidi}^{\mathcal{R}_{Con \to Far}} \leftarrow \mathcal{R}_{Con \to Far}$; // Set rating score
8 $SC_{Bidi}^{\mathcal{R}_{Con \to Del}} \leftarrow \mathcal{R}_{Con \to Del}$; // Set rating score
Output: SC_{Bidi}

To provide stakeholders with the possibility of evaluating service quality, as previously discussed in Section 3.1, another function is defined in the smart contract to make satisfaction scores immutable. This function is detailed in Algorithm 4, allowing consumers, farmers, and deliverers to evaluate each other through a rating score denoted by $R_{User1 \to User2}$ (cf. Table 2). These rating scores refer to the reputation/satisfaction level related to a given FarMarketplace's stakeholder. Note that the functions used for computing these rating scores are not included in the scope of this paper.

3.3. Ecosystem-related key performance indicators

As discussed in Section 2, only limited interactions between blockchain- and infrastructure-related parameters are formalized

[9] The 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.

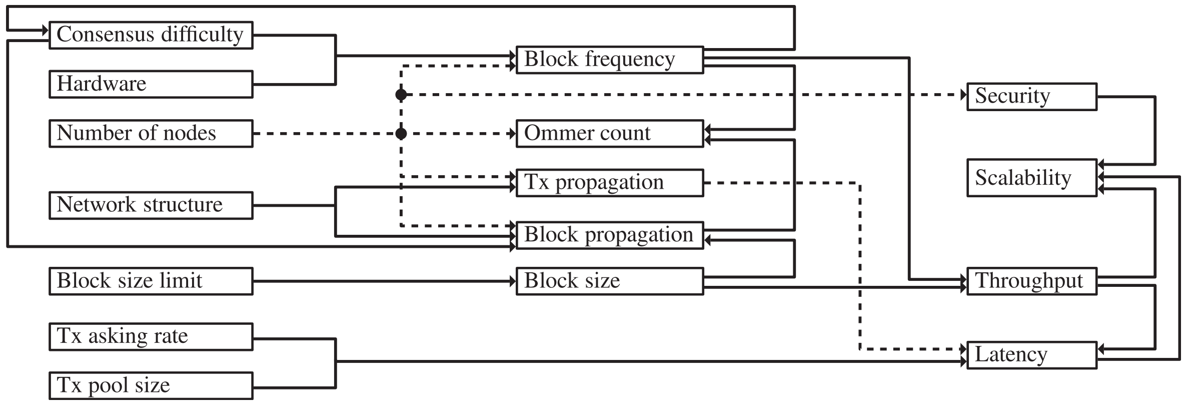


Fig. 5. Performance interaction models regarding blockchain- and infrastructure-related parameters (note, that there is no difference between the solid and dashed arrows, they are used for figure clarity only).

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 Fig. 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 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 Fig. 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 Fig. 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 integrating the set of smart contracts[10].

An overview of the experimental methodology is displayed in Fig. 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

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

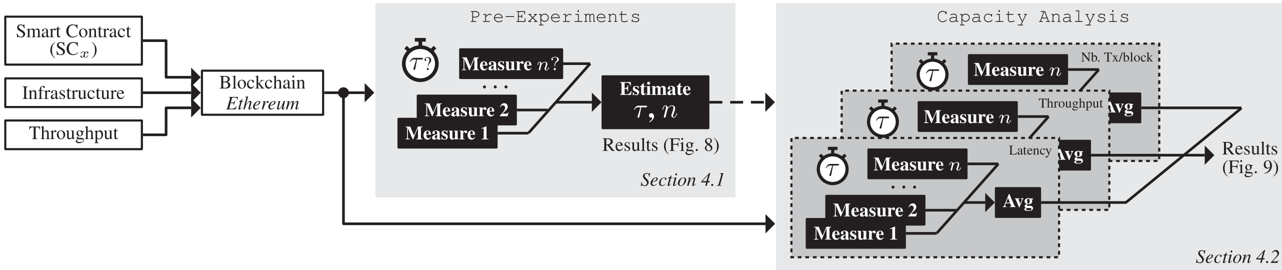


Fig. 6. Methodology for fixing design of experiments.

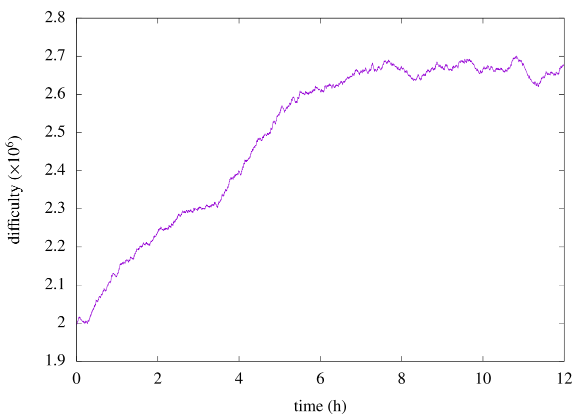


Fig. 7. (Mining) difficulty per timestamp blocks.

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.

Fig. 7 highlights the long-term evolution (over a 12 h 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 s (Pierro, 2019).

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.*, Fig. 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 to be reproducible. To complete the genesis block, the gas limit, which as we know influences the block size, is set arbitrarily to 16 777 216 gas. Furthermore, to ensure that our experimental platform would demonstrate the expected behavior in terms of difficulty ($\#_{total}$) and the influence of the number of threads, a preliminary experimental analysis was performed, as presented in Appendix A.

Based on this configuration setting, pre-experiments were performed to determine the number (n) and duration (τ) of the experiments to be reproduced in the second evaluation stage. As mentioned previously, the latency and throughput offered by the core chain were considered the performance metrics (both being obtained by comparing the timestamps when transactions were generated for a contract and submitted to the chain). Fig. 8 displays the evolution of both metrics, in addition to the number of transactions per block. Note that the experiments were performed after a 30 min block generation period to ensure that stability of the difficulty was achieved.

Fig. 8(a) provides insight into the evolution of the number of transactions per block for 10 min, for ten experiments, one transaction being submitted per second. It can be observed that the convergence is relatively fast and the blockchain remains (reasonably) stable. This is also confirmed by observing the QoS metrics, namely the latency (see Fig. 8(e)) and throughput (see Fig. 8(c)). Throughput is computed as the number of transactions for a block divided by the delay to mine that block; latency is based on the time difference between the emission and validation of the transaction. After 10 min, the latency is in the expected range defined earlier (with a standard deviation less than 1 s and an average mining delay of 11.6 s per block). These experiments allowed us to select a simulation duration of $\tau = 10$ min for the second experimental evaluation stage (a sufficient number of samples being available, 52 blocks on average). In a further step, the confidence was analyzed by considering a greater number of experiments.

Fig. 8(b) displays the evolution of the average number of transactions per block when experiments were added. The relative error corresponds to the difference between the average for a given number and average for 30 experiments. Fig. 8(d) and (f) provide insight into the same analysis for the throughput and latency metrics, respectively. It can be observed that the steady state is achieved with ten experiments and that the increase in the number of experiments does not significantly change the precision. Consequently, in the 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 emission of smart contracts. It is, then, important to identify the capacity offered by the chain to store contract-related information. In this respect, the experiments are

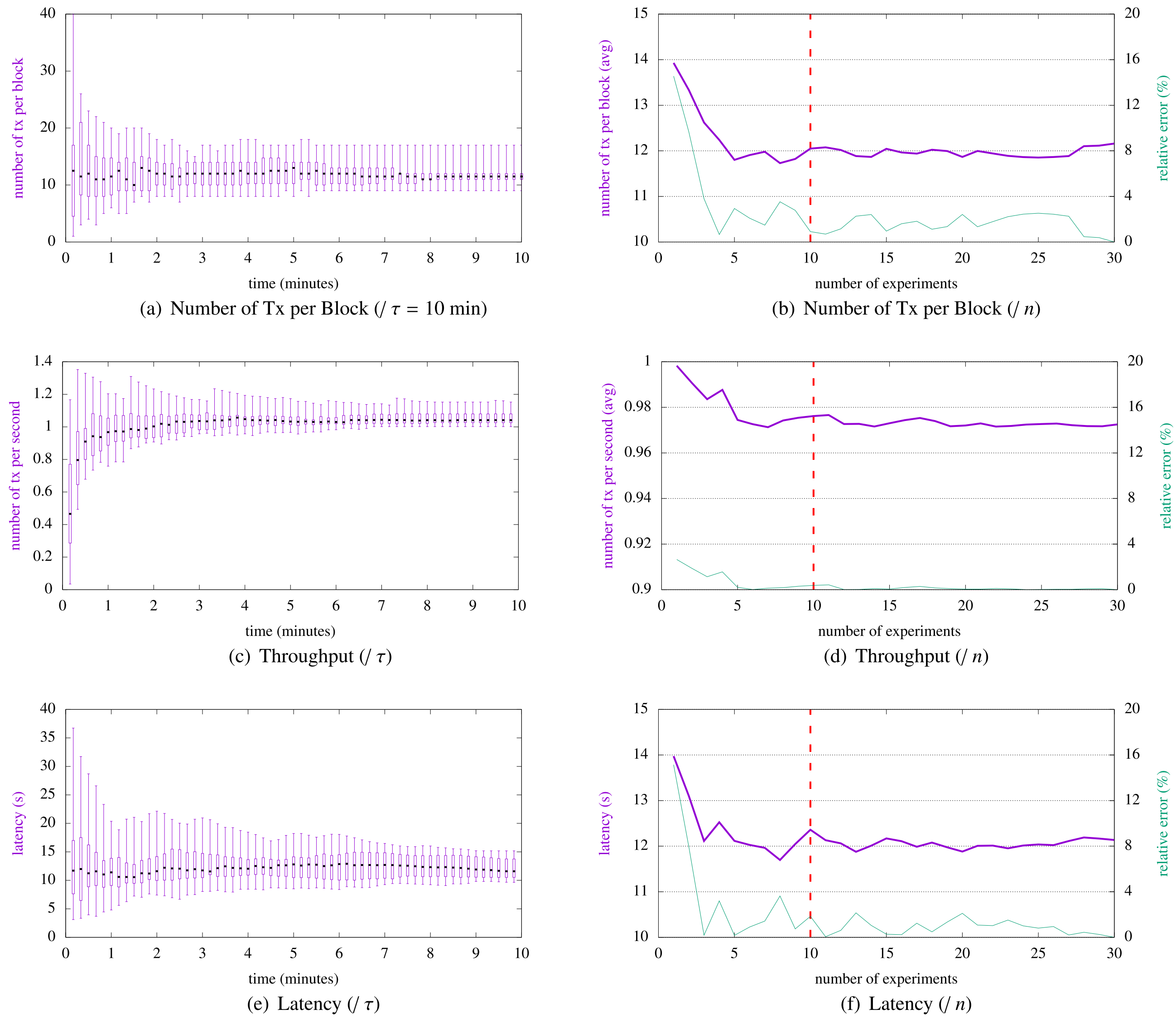


Fig. 8. Evolution of QoS metrics for different number of experiments (n) and in time (τ).

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

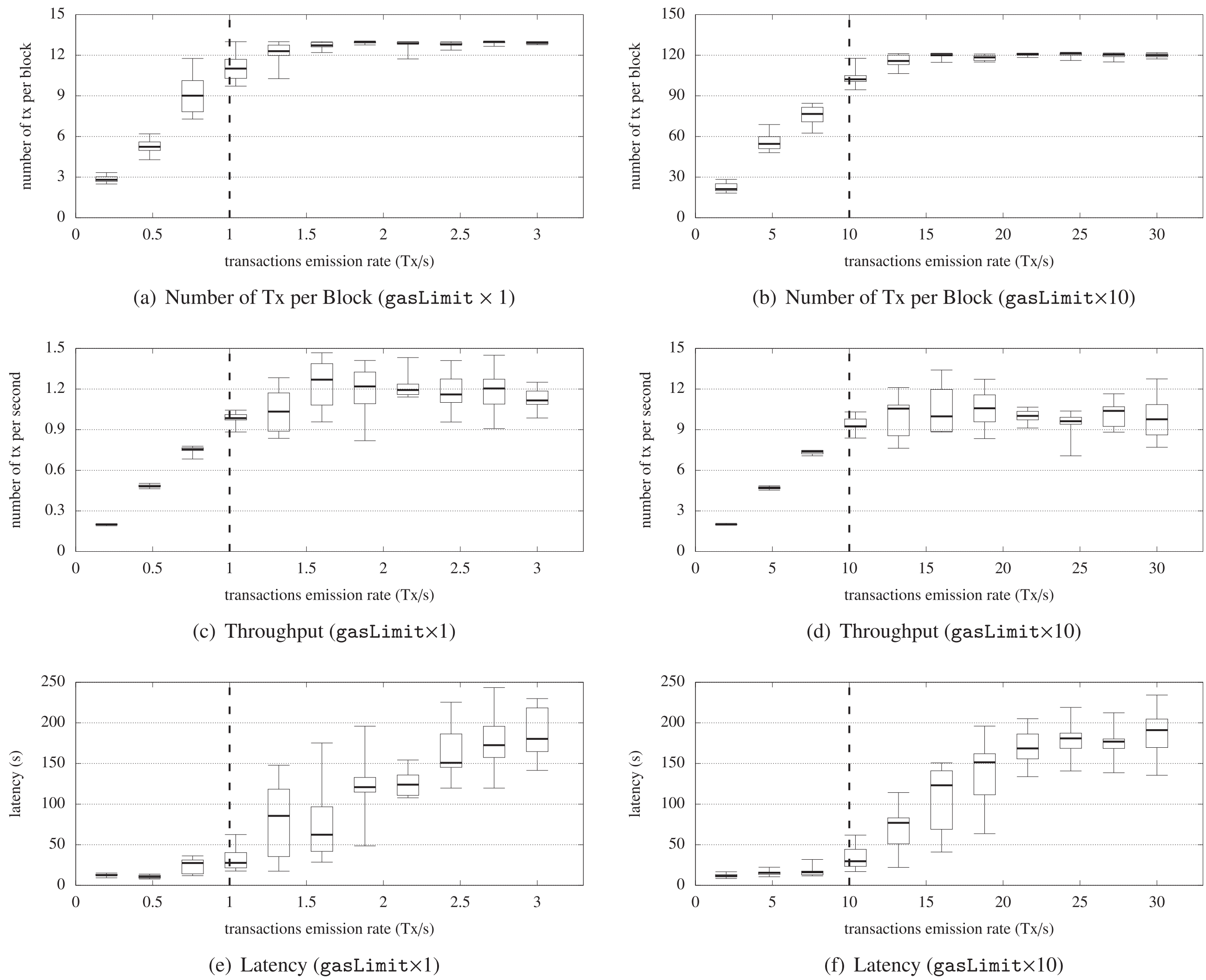
Fig. 9 displays the evolution of the three metrics previously identified; however, this time for a given transaction-emission rate ranging from 1 Tx/s (as previous) to 21 Tx/s. Each point consists of experiments of 10 min, repeated ten times. As the `gasLimit` was arbitrarily chosen (1 blockSize = 16 777 216 gas), we repeated the same set of experiments on another chain with a ten times greater `gasLimit`. Fig. 9(d), (b), and 9(f) correspond to this second experiment.

4.2.1. Number of transactions per block

Fig. 9(a) and (b) provide insight into the evolution of the number of transactions inside a block according to the transactions emission rate. Two behaviors 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 s) 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 transactions, leading to an approximately constant number of transactions per block, with minimal variation between the two experiments.

4.2.2. Latency

Fig. 9(e) and (f) display the average latency of the transactions according to the throughput. We can again extract two behaviors: before and after the *capacity*. With an emission rate less than the *capacity*, as the block is not filled, the transaction is validated and inserted 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 is less than the emission rate, the transaction is queued, leading to a significant increase in the latency.



(a) Number of Tx per Block (`gasLimit` × 1)

(b) Number of Tx per Block (`gasLimit`×10)

(c) Throughput (`gasLimit`×1)

(d) Throughput (`gasLimit`×10)

(e) Latency (`gasLimit`×1)

(f) Latency (`gasLimit`×10)

Fig. 9. Experimental analysis of *capacity* of blockchain.

4.2.3. Throughput

Fig. 9(c) and (d) provide insight into the evolution of the validation throughput. In the first part, the validated throughput is equal to the emission rate because every emitted transaction is validated (leading to the identity function, with minimal differences between the two experiments). In the second part, blocks are saturated such that the validated throughput is limited to a constant, which corresponds to the *capacity* and is defined by the ratio between the number of transactions per block and the delay of mining a block.

As assumed, the number of transactions contained in a block is proportional to the size of the block. By increasing the block size by a factor of 10, we also increase the number of transactions in a block by a factor of 10. The maximal validated throughput, i.e., the *capacity*, is also proportional to the block size.

4.3. Discussions

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

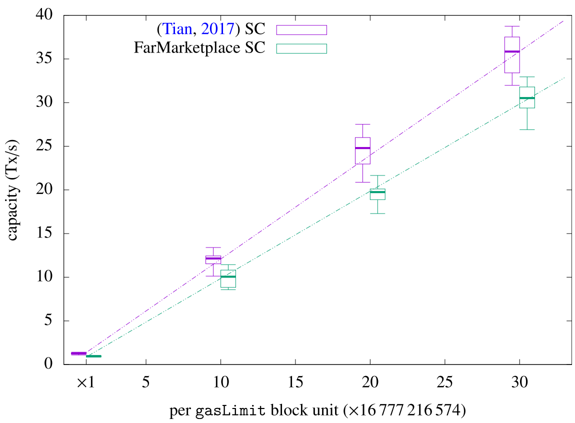


Fig. 10. Comparison of *capacity* evolution for two smart contracts.

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 appearance 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, 2014). 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 s 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 s (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 (2014) (approximately 0, 1%). For 10 min 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.

Thirdly, it contributes to the software development community. To the best of our knowledge, there is only limited research work that thoroughly discusses the interactions between blockchain- and infrastructure-related parameters, and how they influence the overall (end-to-end) QoS performance. The experimental evaluation of the maximum achievable throughput (Tx/s) in the long run (i.e., *capacity*) is a contribution of this research work.

5.3. Limitations and future research directions

Several limitations of our work should be addressed and discussed. The first limitations, related to our experiments, were identified and discussed in Section 4.4; therefore, we refer the reader to that section for experiment-related limitations).

A second limitation relates to the proposed smart contract templates, and particularly to the fact that the set of data items considered in our templates could possibly not cover all the trading requirements for the different types of agricultural goods/markets to be sold/purchased. Even though the Description parameter introduced as part of our smart contracts is sufficiently generic to be extended with any new information that the farmer/seller could deem as relevant (only the hash of the Description is added to the smart contract), it would be convenient to adopt standardized metadata for describing agricultural goods for enhanced interoperability. Semantic- or ontology-based approaches could be investigated and combined with blockchain-based farming ecosystems (Bacco et al., 2019; Lokers et al., 2016).

Although beyond the scope of this research, one key challenge lies in the adoption of blockchain-based solutions by small and medium businesses. The reason for this is twofold: (i) they frequently lack the expertise to invest in blockchain (a common argument that can be found in the literature is that there is no significant adoption of blockchain technology outside of cryptocurrencies) and (ii) clear feedback on the experience gained from the deployment of blockchain-based systems is limited owing to its recent emergence, although selected reports have provided predictions on the potential gains; see, e.g., IBM report[11] that states that blockchain can reduce the time required to trace the source of food from seven days to 2.2 s. It is therefore imperative to make blockchain infrastructures affordable and easy to use in the near future.

Data privacy and security aspects related to blockchain have not been discussed significantly in this paper, although they are of importance in blockchain applications. Indeed, by design, data inserted into a blockchain cannot be erased. Furthermore, the strength of a public blockchain is that everyone can download and verify blocks and transactions, thus leaving room for privacy concerns. Although sensitive data in the proposed solution are stored apart from the blockchain (in an external database called FarMarketDB), the "hash" of that data is added to the blockchain (via smart contract). Other more advanced solutions could be explored in the future, similar to the ones proposed in Kosba et al. (2016); Bünz et al. (2020).

Our study of the parameters' influence is also limited by the chosen blockchain; Ethereum runs on the Geth client. In fact, the diversity of consensus protocols and chain parameters supporting existing blockchain technologies makes it difficult to objectively compare two technologies. To do this effectively, our analysis should be extended to other chains that function with other consensus protocols. With this extension, a similar comparison basis could be defined, enabling a better choice for the blockchain infrastructure selection.

Finally, we must highlight the fact that the relation considered in Section 4.1 between "difficulty" and "Blocktime" is based on an assumption (this assumption being further detailed in A), and further research should be performed to determine more accurate numerical values (e.g., regarding the average difficulty calculation).

CRediT authorship contribution statement

Guilain Leduc: Conceptualization, Methodology, Software, Writing – original draft. **Sylvain Kubler:** Supervision, Writing – review & editing. **Jean-Philippe Georges:** Supervision, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly supported by the French PIA project "Lorraine Université d'Excellence" reference ANR-15-IDEX-04-LUE. We would like to express special thanks to Mario Lezoche for his contribution to the brainstorming challenges and ideas related to blockchain-based farming.

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

Appendix A. Preliminary benchmark analysis

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

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 10 s in Ethereum, this relation is frequently simplified by $d = 10 \times \#_{tot}$ (cf., Section 4.1).

$$\tau = \frac{d}{\#_{tot}} \tag{A.1}$$

This relation can be interpreted as the mean of an exponential distribution of parameter $\lambda = \frac{\#_{tot}}{d}$. Indeed, as the mining of the block is comparable to a brute force attack for a puzzle solution in a set sufficiently large, the search of the solution can be modeled by a continuous memoryless distribution. This model also provides an explanation of the hashrate additivity property, as a network composed of N nodes results in a system of $(n_i)_{i\in N}$ independent nodes seeking the solution to the puzzle, each having a given hashrate denoted by $\#_i$. Therefore, for each node n_i, the random value denoted by X_i for finding a solution follows an exponential parameter distribution $\frac{\#_i}{d}$. The network's random value X_{tot} to find a solution among all network nodes is therefore equal to $X_{tot} = min(\{X_i\}_{i\in N})$ as a solution is found if and only if one node solves the puzzle, following an exponential network distribution $\frac{\sum_{i\in N}\#_i}{d}$. Overall, the hashrate of a network can be determined by summing the hashrate of the entire network.

Appendix A.2. Validity of the model

To verify the consistency of the proposed model and benchmark platform, the relation between blocktime and hashrate was tested. To achieve this, a blockchain with the same initial parameters (including the same difficulty) was performed with a variable number of hashrates. The experiment duration (10 min) was sufficiently short to neglect the change in difficulty due to the feedback control. In the experiment that produced the maximum number of blocks, which is more likely to be influenced by this control, 64 blocks were produced, which could modify the difficulty in Ethereum according to A.3 by up to 3% (this equation is further detailed in the next section). Indeed, according to A.3, a block can modify the difficulty of a block by a factor $\frac{1}{2048}$. Thus, after 64 blocks, $\Delta d = \left|\left(1 \pm \frac{1}{2048}\right)^{64} - 1\right| < 3.2\%$. 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.

$$\tau = \frac{d}{\#_{thread} \times nb_{thread}} \tag{A.2}$$

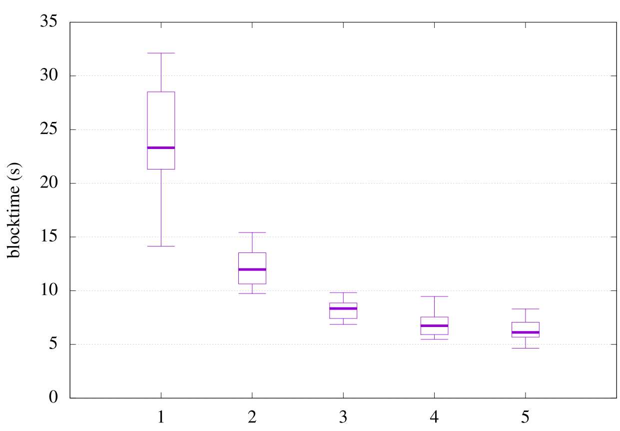


Fig. A.11. Evolution of average blocktime according to number of threads.

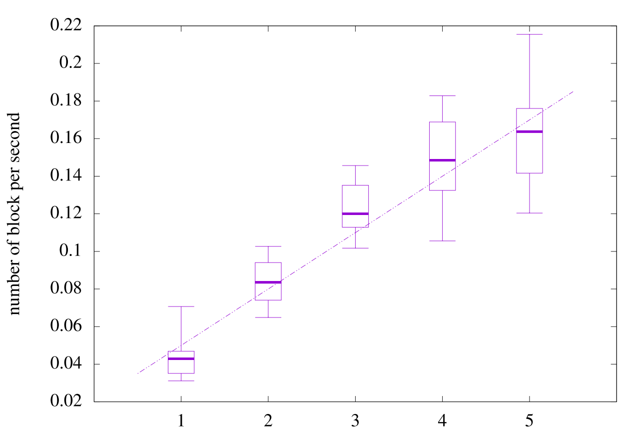


Fig. A.12. Evolution of inverse of average blocktime according to number of threads.

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, 2014), and is given in (A.3) $\forall n > 0$:

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

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

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, *epsilon* 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 (2014): before the difficulty bomb, if no ommer appears, and if the delay between two blocks is between 9 and 18 s, 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

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

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

Albert, E., Correas, J., Gordillo, P., Román-Díez, G., Rubio, A., 2020. GASOL: gas analysis and optimization for Ethereum smart contracts. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Dublin, Ireland, pp. 118–125.

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 Commun. Surv. Tutorials 21 (2), 1676–1717.

Awan, S.H., Ahmed, S., Safwan, N., Najam, Z., Hashim, M.Z., Safdar, T., 2019. Role of Internet of things (IoT) with blockchain technology for the development of smart farming. J. Mech. Contin. Math. Sci. 14, 170–188.

Bacco, M., Barsocchi, P., Ferro, E., Gotta, A., Ruggeri, M., 2019. The digitisation of agriculture: a survey of research activities on smart farming. Array 3, 100009.

Bermeo-Almeida, O., Cardenas-Rodriguez, M., Samaniego-Cobo, T., Ferruzola-Gḿez, 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, pp. 44–56.

Bez, Mirko, Fornari, Giacomo, Vardanega, Tullio, 2019. The scalability challenge of ethereum: An initial quantitative analysis. International Conference on Service-Oriented System Engineering. IEEE, pp. 167–176.

Blandford, D., 2019. Supply response to increased demand for food. In: Global Challenges For Future Food And Agricultural Policies, vol. 1, p. 61.

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.

Bumblauskas, D., Mann, A., Dugan, B., Rittmer, J., 2020. A blockchain use case in food distribution: do you know where your food has been? Int. J. Inf. Manag. 52, 102008.

Bünz, Benedikt, Agrawal, Shashank, Zamani, Mahdi, Boneh, Dan, 2020. Zether: towards Privacy in a Smart Contract World. Financial Cryptography and Data Security. Springer International Publishing, pp. 423–443.

Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, 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, pp. 1–4.

Davcev, D., Kocarev, L., Carbone, A., Stankovski, V., Mitresk, 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, pp. 51–58.

Decker, C., Wattenhofer, R., 2013. Information propagation in the bitcoin network. In: IEEE International Conference on Peer-To-Peer Computing. Trento, Italy,

September, 1–10.
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, pp. 7–19.
Di Francesco Maesa, Damiano, Mori, Paolo, Ricci, Laura, 2017. Blockchain Based Access Control. Distributed Applications and Interoperable Systems, 10320. Springer International Publishing, pp. 206–220.
Dujak, D., Sajter, D., 2019. Blockchain applications in supply chain. In: SMART Supply Network. Springer, Cham, pp. 21–46.
Esposito, Christian, Ficco, Massimo, Gupta, Brij Bhooshan, 2021. Blockchain-based authentication and authorization for smart city applications. Inf. Process. Manag. 58 (2).
Fan, Caixiang, Ghaemi, Sara, Khazaei, Hamzeh, Musilek, Petr, 2020. Performance evaluation of blockchain systems: a systematic survey. IEEE Access 8, 126927–126950.
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. J. Clean. Prod. 121031.
Galvez, J.F., Mejuto, J.C., Simal-Gandara, J., 2018. Future challenges on the use of blockchain for food traceability analysis. Trac. Trends Anal. Chem. 107, 222–232.
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. J. Clean. Prod. 240, 118021.
Hang, L., Ullah, I., Kim, D.H., 2020. A secure fish farm platform based on blockchain for agriculture data integrity. Comput. Electron. Agric. 170, 105251.
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, pp. 97–101.
Iqbal, R., Butt, T.A., 2020. Safe farming as a service of blockchain-based supply chain management for improved transparency. Cluster Comput. 1–12.
Juma, H., Shaalan, K., Kamel, I., 2019. A survey on using blockchain in trade supply chain solutions. IEEE Access 7, 184115–184132.
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. Int. J. Prod. Econ. 219, 179–194.
Kamilaris, A., Fonts, A., Prenafeta-Boldú, F.X., 2019. The rise of blockchain technology in agriculture and food supply chains. Trends Food Sci. Technol. 91, 640–652.
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, p. 100315.
Kosba, Ahmed, Miller, Andrew, Shi, Elaine, Wen, Zikai, Papamanthou, Charalampos, 2016. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. Symposium on Security and Privacy. IEEE, pp. 839–858.
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. J. Clean. Prod. 252, 119902.
Kumar, M.V., Iyengar, N.C.S., 2017. A framework for Blockchain technology in rice supply chain management. Adv. Sci. Technol. Lett. 146, 125–130.
Laux, C.M., Hurburgh Jr., C.R., 2012. Using quality management systems for food traceability. J. Ind. Technol. 26 (3), 1.
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 Generat. Comput. Syst. 86, 641–649.
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. Comput. Ind. 116, 103187.
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.
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. Canada, Vancouver, pp. 1–6.
Lokers, R., Knapen, R., Janssen, S., van Randen, Y., Jansen, J., 2016. Analysis of Big Data technologies for use in agro-environmental science. Environ. Model. Software 84, 494–504.
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.
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, pp. 1–10.
Mao, D., Hao, Z., Wang, F., Li, H., 2019. Novel automatic food trading system using consortium blockchain. Arabian J. Sci. Eng. 44 (4), 3439–3455.
Nowiński, W., Kozma, M., 2017. How can blockchain technology disrupt the existing business models? Entrep. Bus. Econ. Rev. 5 (3), 173–188.
Panarello, A., Tapas, N., Merlino, G., Longo, F., Puliafito, A., 2018. Blockchain and IoT integration: a systematic survey. Sensors 18 (8), 2575.
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.
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, pp. 1162–1167.
Pierro, G.A., Rocha, H., 2019. The influence factors on ethereum transaction fees. In: 2nd IEEE/ACM International Workshop on Emerging Trends in Software Engineering for Blockchain. Canada, Montreal, pp. 24–31. May.
Pinna, A., Ibba, S., 2018. A blockchain-based Decentralized System for proper handling of temporary Employment contracts. In: Science and Information Conference. Springer, Cham, pp. 1231–1243.
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.
Rabah, K., 2018. Convergence of AI, IoT, big data and blockchain: a review. Lake Inst. J. 1 (1), 1–18.
Ramli, Carroline Dewi Puspa Kencana, Nielson, Hanne Riis, Nielson, Flemming, 2014. The logic of XACML. Sci. Comput. Program. 83, 80–105.
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, pp. 887–896.
Rose, D.C., Chilvers, J., 2018. Agriculture 4.0: broadening responsible innovation in an era of smart farming. Front. Sustain. Food Syst. 2, 87.
Sajana, P., Sindhu, M., Sethumadhavan, M., 2018. On blockchain applications: Hyperledger fabric and Ethereum. Int. J. Pure Appl. Math. 118 (18), 2965–2970.
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. July, Atlanta, USA, pp. 546–551.
Singh, Nikita, Vardhan, Manu, 2019. Distributed ledger technology based property transaction system with support for IoT devices. Int. J. Cloud Appl. Comput. 9 (2), 60–78.
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.
Sumathi, M., Sangeetha, S., 2020. Blockchain based sensitive attribute storage and access monitoring in banking system. Int. J. Cloud Appl. Comput. 10 (2), 77–92.
Surasak, T., Wattanavichean, N., Preuksakarn, C., Huang, S.C.H., 2019. Thai Agriculture Products Traceability System Using Blockchain and Internet of Things, p. 15. Editorial Preface From the Desk of Managing Editor, 10(9).
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. Transforming Climate Finance and Green Investment with Blockchains. Academic Press, pp. 137–152.
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, pp. 1–6.
Tripoli, M., Schmidhuber, J., 2018. Emerging opportunities for the application of blockchain in the agri-food industry. In: FAO and ICTSD: Rome and Geneva, vol. 3. Licence: CC BY-NC-SA.
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, pp. 1357–1361.
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, pp. 137–144.
Vermesan, O., Friess, P., 2016. Digitising the Industry-Internet of Things Connecting the Physical, Digital and Virtual Worlds. River Publishers.
Vukolić, M., 2015. The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: International Workshop on Open Problems in Network Security. Zurich, Switzerland, October, pp. 112–125.
Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.J., 2017. Big data in smart farming – a review. Agric. Syst. 153, 69–80.
Wood, G., 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger, pp. 1–32. Ethereum project yellow paper, 151(2014).
Xiao, Y., Zhang, N., Lou, W., Hou, Y.T., 2020. A survey of distributed consensus protocols for blockchain networks. IEEE Commun. Surv. Tutorials. https://doi.org/10.1109/COMST.2020.2969706.
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, pp. 45–50.
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 Commun. Surv. Tutorials 21 (3), 2794–2830.
Xu, X., Weber, I., Staples, M., 2019. Case study: AgriDigital. In: Architecture for Blockchain Applications. Springer, Cham, pp. 239–255.
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. Comput. Ind. 109, 83–99.
Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. 14 (4), 352–375.