REA, Triple-Entry Accounting and Blockchain:

Converging Paths to Shared Ledger Systems

Current version: May 4, 2021

Juan Ignacio Ibañez[1]

Chris N. Bayer[2]

Paolo Tasca[3]

Jiahua Xu[4]

ABSTRACT

The concept of shared ledger systems offering a single source of truth has repeatedly called traditional bookkeeping into question. Improving upon the long-standing double-entry systems, solutions such as the Resource-Event-Agent (REA) accounting framework, triple-entry accounting (TEA) and blockchain have been advanced. However, to date, the historical development of these concepts remains murky. This paper conducts a genealogical analysis of shared ledger systems, in particular tracing the development of REA, TEA and blockchain. We show how the REA framework has had a distinct influence on independent streams of research in the field of TEA, and how this interaction may be traced to the present incarnation of shared ledger systems in blockchain. In doing so, we duly acknowledge the influence of key individuals contributing to this development, correct common misconceptions and map out how the paths of REA, TEA and blockchain overlap in the realm of shared ledger systems.

Keywords: triple-entry accounting, REA accounting model, single source of truth, blockchain, distributed ledger technology.

---

[1] Centre for Blockchain Technologies, University College of London.

[2] Centre for Blockchain Technologies, University College of London.

[3] Centre for Blockchain Technologies, University College of London.

[4] Centre for Blockchain Technologies, University College of London.

This paper also benefited from comments by a number of individuals per interview or per written correspondence. We would like to gratefully acknowledge Chris Cook, Chris Odom, Craig S. Wright, David Hartley, G. Ken Holman, Ian Grigg, Marcos Miranda, Melisa Baudon, Robert Haugen, Todd Boyle and Vincent Piscaer. Any mistakes are our own.

I. INTRODUCTION

Scientific revolutions begin with extraordinary investigations, through which scientists challenge existing paradigms (Kuhn 1996). In the accounting profession, one set of such investigations took place between 1458 and 1494, represented by the works of Benedetto Cotrugli (1573), Marino de Raphaeli (1475) and Luca Pacioli (1494). These investigations resulted in the development of the double-entry accounting method, which replaced the single-entry system in effect at the time, and was widely adopted throughout the five centuries that followed.

The 1970s saw the beginning of a second accounting revolution in the form of shared ledger systems. Although this novel concept does not necessarily change the fundamental philosophy of accounting, it challenges the gold standard status of traditional double-entry accounting (Boyle 2003f; Grigg 2005b; McCarthy 1982, 2001). Introducing a landscape of new possibilities for accounting, shared ledger systems have been bringing about a diverse set of potential opportunities for application, ranging from external and internal integration to intrinsic reconciliation, invoice automation, dispute resolution, anti-corruption, and more (Alawadhi, et al. 2015; Boyle 2002, 2003f; Dai 2017; Dai and Vasarhelyi 2017; ICAEW 2018; Mainelli and Smith 2015; Mohanty 2018, 47; Request 2018a, 2018b).

In recent years, the computer science community has been developing various shared ledger systems underpinned by blockchain technology (Vijai, Elayaraja, Suriyalakshimi, and Joyce 2019). Meanwhile, shared ledger systems advanced concurrently by the accounting community have inspired several solutions involving real-time auditing and continuous assurance (Dai 2017; Dai and Vasarhelyi 2017; Auditchain 2018; see also Alawadhi et al. 2015).

These technological advances, however, have yet to be academically parsed, resulting in historiographical gaps (Cai 2019, 3). This is especially the case for a particular kind of shared ledger design known as "triple-entry accounting" or TEA. Shared ledger systems in general, and TEA in particular, are widely believed to have potentially sweeping implications for accounting, resulting from radical cryptographical developments. (Rao 2020; see also Cai 2019; Gröblacher and Mizdraković 2019; Inghirami 2019; Pacio 2018a). However, the reverse is also true.

In this paper, we posit the existence of an opposite chain of causality: a revolution in *accounting* broadened the applicability of cryptography. Specifically, we claim that the research of William E. McCarthy (1982), Todd Boyle (2000a, 2000b, 2000c), Ian Grigg (2005b) and, possibly, Satoshi Nakamoto (2008), constitute an interrelated set of works telling the cohesive story of shared ledger systems.

Moreover, we show that TEA is, in part, a historical byproduct of the Resource-Event-Agent (REA) accounting framework designed by McCarthy. While parallels between TEA and the REA framework have been noted (A. Gomaa et al. 2019; Grigg 2017a, 2017b, 2017c), the historical influence of the latter over the former remains overlooked as a result of an underappreciation of the stream of work revealing the "missing link" between REA and TEA, primarily carried out by Todd Boyle (2000a, 2000b, 2000c).

We seek to fill in these gaps by conducting a genealogical analysis of shared ledger systems, correcting common misconceptions, and investigating the largely unacknowledged influence of William E. McCarthy on TEA and today's blockchain technology.

To this end, we perform a comprehensive literature review that covers: William E. McCarthy (1982), Robert Haugen (Haugen and McCarthy, 2000), Todd Boyle (2000a, 2000b, 2000c), Ian

Grigg (2005b), G. Ken Holman (2019), Yuji Ijiri (1975, 1982) and Chris Cook (2002). We place a particular emphasis on the works of Todd Boyle, the missing link between REA and TEA. We also interviewed pioneers in REA, TEA and blockchain to document the oral history of shared ledger systems. We find that the current explosion of shared ledger system use cases result from the convergence of three streams of research, developing in parallel and occasionally interacting with each other.

The rest of the paper is structured as follows. First, we elucidate the streams of accounting innovation at issue, comparing them, clarifying the terminology and defining the essential concepts. Next, we discuss the historical development of these shared ledger systems before concluding with final remarks.

II. Revisiting Accounting: Three Streams of Innovation

The REA Framework

Resource-Event-Agent (REA) is a generalised accounting framework that enables a computer software model for an Enterprise Information System (Haugen and McCarthy 2000; McGeerts and McCarthy 2006) originally conceived by William E. McCarthy (1982, 2001). At the core of this model lies the representation of transactions as business *events*, where the companies' *agents* exchange *resources*.

In a nutshell, REA is a modelling tool for a centrally defined database containing atomic transaction records with all relevant variables. It proposes semantic abstractions generalising business events (Boyle 2000c) that entail a "set of classes, relationships, and functions in a universe of discourse" (Haugen and McCarthy 2000). REA overcomes several challenges in the usage of existing accounting models, including imperfect classification schemes, crude bookkeeping data, and the lack of integration of the accounting system with non-accounting information systems in an enterprise. While REA supports the reporting artefacts of double-entry accounting such as balance sheets and income statements (Gal and McCarthy 1986; McCarthy 2001), the model purports to replace the classical double-entry accounting system with an information system integrated to all functional areas of an enterprise, i.e. not just limited to the accounting department.

Originally, McCarthy and his followers were spelling out the conceptual framework for an integrated business information system for the various areas of a *single* company. Nonetheless, with the advent of the Internet, the REA model was extended to multiple business entities in a trading community, that is, to *inter-company accounting*: a shared ledger system. For instance, Haugen and McCarthy (2000) extended the REA model to create an event-driven generalised representation of material flows supporting a single source of truth throughout supply chains and demand chains.

Traditionally, intercompany transactions are recorded twice, separately from the perspective of each transacting party (ISO/IEC 2015, 3). For instance, a sale recorded into account receivables by a seller is mirrored by the buyer recording the same transaction into accounts payable (Boyle 2000f). Thus, accounting records are viewpoint-dependent. McCarthy, however, proposed a representation of real-world business events from a viewpoint-independent, inter-enterprise perspective (ISO/IEC 2015, vii). This proposal was designed with the ANSI/X3/SPARC architecture for a Database Management System in mind (McCarthy 1982, 557; see also Tsichritzis and Klug, 1978) and eventually resulted in the Open-edi Distributed Business Transaction

Repository (OeDBTR) project led by the ISO/IEC JTC 1/SC 32/WG 1 eBusiness Working Group, which is discussed later in the section *Differences and Similarities Between REA, Triple-entry Systems and Blockchain*.

Triple-Entry Accounting

Triple-entry accounting is a type of shared ledger system with a three-pronged consensus mechanism, resulting from the parallel yet independent work of Ian Grigg (2000, 2005a, 2005b) and Todd Boyle (2001b, 2001g). TEA is based on a shared transaction repository (STR) and relies on triple-signed receipts (Odom 2013, 2015) to reach an agreement on the record.

In order to update the shared record with a new transaction record, two parties need to be involved: one initiates a transaction entry – called “request”, “offer”, or “transaction draft” – over the STR. The STR verifies the transaction, creates a receipt, and posts it in a shared database, before forwarding it to the parties. Note that it is possible for transaction counterparties to receive multiple copies of a receipt, as in the event of a recovery from system interruption, the STR reads back the database's log and re-forwards the receipt. The other party then accepts the entry.

This procedure can be understood as a signature-gathering process: one party adds their signature to the transaction entry draft, and the counterparty accepts by countersigning before the entry is processed by the STR, which can be a middleware server or a distributed ledger. The STR checks the validity of the signatures and then, if everything is in order, signs off on the entry (Boyle 2000e, 2001g; Grigg 2005b). This mechanism generates a hashed triple-signed receipt, such that all the parties hold the same data that cannot be manipulated or lost: a single, shared entry serving as the *single source of truth*. In computer science terms, this is an implementation of the WYSIWIS ("What You See Is What I See") principle (Brown 2020).

In TEA, a local copy of the shared transaction repository may be integrated as a subledger to the general ledger of transactional parties. However, because the shared entry is the sole reliable source of the transaction record, some call this methodology "single-entry bookkeeping" (Pacio 2018a, 2020). Nevertheless, we discourage usage of this term. For one, the term "single-entry bookkeeping" is already reserved for simplified bookkeeping systems in differentiation from more complex double-entry bookkeeping. Historically, basic single-entry systems only record stock accounts, i.e. assets and liabilities, without including flow accounts such as revenues and expenses (Ijiri 1986, 746) that double-entry bookkeeping systems do, and without two entries or sides (debits and credits) for each transaction (Grigg 2005b). Unlike simplified, single-entry bookkeeping, TEA is compatible with double-entry representations (Boyle 2000c, 2000a, 2000b). In effect, Ian Grigg (2005b) conceived TEA to consist of pairs of double entries with each pair connected to a central receipt, resulting in three parties holding the triple-signed receipts. This interconnection is another reason to refer to this concept as triple-entry, rather than single-entry. Following this logic, triple-entry accounting has become an established term in the industry (Gröblacher and Mizdraković 2019).

At this point, we wish to clarify the polysemy of the term "entry". In the context of "historical" single-entry bookkeeping, an "entry" is a record of a change in stocks accounts such as assets and liabilities (Ijiri 1986, 746) without a counterpart to that record. In "modern" single-entry bookkeeping, it is the record of income or expenses, also without a counterpart (IRS 2015). In double-entry bookkeeping, an entry is a debit or a credit record (IRS 2015).[5] In McCarthy's REA and Boyle's (2001g, 2003d) STR, an entry is an atomic record of an economic event that does not need balancing. In Boyle's (2003d) TEA, the three entries are the STR entry and two (optional) private transaction stubs[6] for the parties (this is termed a "stub – shared entry – stub" structure).

Finally, in Grigg's TEA, the entries are the three signature records: the three signed messages of the parties (Grigg 2005), which correspond with Boyle's (2001b) offer, acceptance and validation. Furthermore, the single copy of the triple-signed record is in three places (Grigg 2005). Grigg's concept of "triple-entry" is widespread in the blockchain world, with his definition of TEA trumping Boyle's. Since Grigg's idea of "entry" is different from the common accounting acceptation of "entry", this all means that TEA's triple recording of *signatures* does not necessarily challenge the double-entry's bilateral recording of *transactions*.[7]

Blockchain

Distributed ledger technology (DLT) is a recent proposal for a shared ledger system. In a DLT system, the record of transactions is collectively maintained by a network of computers/nodes, which do not necessarily trust each other. The trustless nature of the network requires the participating nodes to resort to a pre-determined protocol that specifies the transaction logging and verification process to achieve the consensus on (the single set of) shared records (Brown 2016).

---

[5] For a discussion of the different views on the specificity of double-entry bookkeeping, see Goldberg (1965, 215-219).

[6] A stub is the counterfoil of a transaction receipt. As envisioned by Boyle (2001b), parties may optionally insert non-essential data such as a memo in them.

[7] For an argument of why it is nevertheless *desirable* to discard the double-entry model at the bookkeeping level in shared ledger systems, see Boyle (2000c, 2001f, 2003f) and McCarthy (2001).

Blockchain is an architectural configuration of DLT based on chains of blocks. A block is a timestamped package containing a series of transaction data as well as a cryptographic hash of the previous block. The resultant record sequences are thus cryptographically appended into the form of a "chain" (Perez, Xu, and Livshits 2020).

The Bitcoin blockchain is commonly deemed the first historical example of a DLT system. However, not all DLT networks are blockchains (e.g. R3's Corda; Mohanty 2019). Non-blockchain DLT systems are nevertheless inspired by blockchain technology. For this reason, "blockchain" is customarily used also to encompass the larger family of distributed ledger technologies, i.e. community consensus-based ledgers, whose constituent data may or may not form into chains of blocks.[8] We use this broad definition of "blockchain" in the remainder of this paper.

While Bitcoin is often described as an example of a triple-entry bookkeeping system (Grigg 2011, see also Tyra 2014 and February 2020a, 2020b) because it relies on signed messages resembling a three-signature model, two particularities are worth noting. Firstly, the third signature is not provided by a single operator, but by the network of participating nodes (Zahnentferner 2018). Specifically, a miner[9] signs off on a valid transaction by means of including it in a new block; the community accepts the block by including the hash of this block in the next block, forming a "chain." Miners are incentivised to conduct transaction verification as for each block mined, they receive a block reward together with fees collected from all the transactions included in the block through a so-called "coinbase transaction." In that sense, the coinbase transaction of the block,

[8] For further understanding of the family of the distributed ledger technologies see Tasca and Tessone (2019).
[9] Miners concatenate verified transactions to form a block, and then compete to obtain the solution, termed "nonce", to a cryptographic challenge posed by the blockchain protocol. The winning miner broadcasts this new block of transactions together with the nonce to the network, extending the "chain."

together with community consensus, act as a manifestation of the third-party validation of the transaction.

Secondly, Bitcoin is a *payment* system, and payment, in contrast to exchange, is a one-way transaction where the receiver's confirmation (the *second* signature) is not always needed.[10] However, the receiver expresses acknowledgement of the occurrence of the transaction in the act of later spending the amount received.

This acknowledgement arises from Bitcoin following the UTXO (Unspent Transaction Output) model, in which a transaction input (amount to be sent by the payer) must at least be equal to the transaction output (amount to be received by the payee). When that transaction input exceeds the output, the unspent amount, namely the difference between the input and the output[11] is returned to the payer as change (unspent output for the payer). The amount transferred, on the other hand, becomes a new unspent amount for the payee, which must be entered in its *entirety* as an input for a later transaction. Therefore, the UTXO model necessarily requires the new payer (and former payee) to sign off on a message that confirms receipt of a prior payment output in its totality (or of a mining reward which resulted in the creation of new bitcoins through a coinbase transaction). Hence, Bitcoin can be considered a triple-entry system, with the second signature[12] being placed asynchronously.

However, note that not all blockchains follow a triple-entry model. For instance, the data language of Ethereum mainly consists of database tables. Thus, although Ethereum involves running

---

[10] However, a second signature can be incorporated into a blockchain system. For example, Open-Transactions (Odom 2015) proposes a cheque scheme to transact bitcoins off-chain that requires three signatures: cheque issuer, cheque receiver and notary.

[11] And the eventual transaction fees, if applicable.

[12] We rely on a broad notion of signature (as any token that attests agreement at some point).

programs through smart contracts and agreeing on the results through the proof-of-work consensus mechanism, there exist no "signed receipts" unless programs with that specific design are included. Therefore, Ethereum itself is not a TEA system, but it may enable TEA platforms and TEA applications within its environment through smart contracts (such as Request, Balanc3, PayPie and Ledgerium).

Differences and Similarities Between REA, Triple-entry Systems and Blockchain

REA is a generalised framework establishing an ontology, whereas TEA can be deemed an implementation of REA. Nevertheless, when applied specifically to inter-entity transactions, REA offers a concept that is functionally equivalent to TEA: The Open-edi[13] Distributed Business Transaction Repository or OeDBTR (ISO/IEC 2015; McCarthy and Holman 2019).

OeDBTR is the term assigned, within the REA ontology, to a system that tracks the immutable history of events triggering changes of state in multiple business entities, relying on the independent view of the transaction as a single source of truth and the open-edi standard for electronic data interchange described in ISO/IEC 15944-21 (McCarthy and Holman 2019, see also Holman 2019). TEA and the OeDBTR thus share a fundamental characteristic: a viewpoint-independent record of transactions that is shared between two or more parties, and that can support different local "views" of the transactions.

As an example, let us imagine that Alice buys from Bob two bicycles in 2019, which are identical in every possible way: one in March for 70 USD and one in April for 80 USD. In January 2020,

[13] Open-edi is an ISO/IEC standard for electronic data interchange (ISO/IEC 2010).

Alice sells one bicycle to Charlie for 100 USD. In order to record the transactions, Alice proposes transaction entries over an STR. Bob and Charlie accept, and the STR verifies the transactions and enters them in a shared record.

REA, TEA and blockchain address different dimensions of a shared ledger system: database structure, reliability and consensus, respectively. As a consequence, each model is naturally different: an OeDBTR record must follow REA database specifications to achieve semantic expressiveness, a TEA transaction record requires three signatures to attest the accuracy of the information stored, and a blockchain requires a chain of hashed blocks agreed upon by the community. However, precisely because they address different facets of the record, they are *compatible*. Moreover, at a fundamental level, the records are *comparable*: they do not largely differ. A stylised comparison of them illustrating the first transaction in the example above is presented in **Figure 1**.

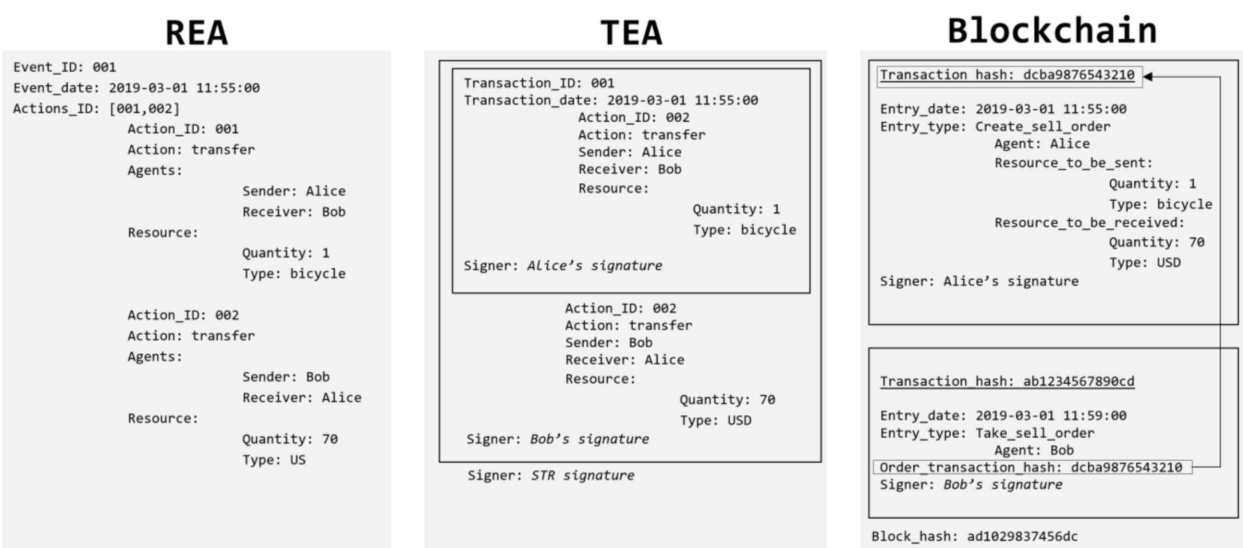

Figure 1: stylised REA, TEA and blockchain transaction records. Inspired by the Valueflows (2017) model for REA, Grigg's (2005b) signed receipt for TEA, and Ethereum smart contracts (see Appendix A: Profit-calculating smart contracts). Note that the third box in this figure depicts a

BLOCKCHAIN RECORD IN A DECENTRALISED EXCHANGE, IN WHICH THE TWO TRANSACTIONS WHERE THE ORDER IS POSTED AND TAKEN, RESPECTIVELY, ARE BOTH RECORDED IN THE SAME BLOCK. HOWEVER, IT IS ALSO POSSIBLE THAT BOTH TRANSACTIONS ARE RECORDED IN DIFFERENT BLOCKS, OR THAT THE TRANSACTION IS THE RESULT OF THE EXECUTION OF A SMART CONTRACT INSTEAD OF AN ORDER ACCEPTANCE IN AN EXCHANGE.

While the transaction record is viewpoint-independent, each of the parties to the system can see it from their particular perspective. How each transaction entry in a shared record manifests itself differently from the viewpoint of each party to the system is best represented as a 3-dimensional bookkeeping grid, by which N sheets equals the N parties to the system (illustrated in **FIGURE 2**).

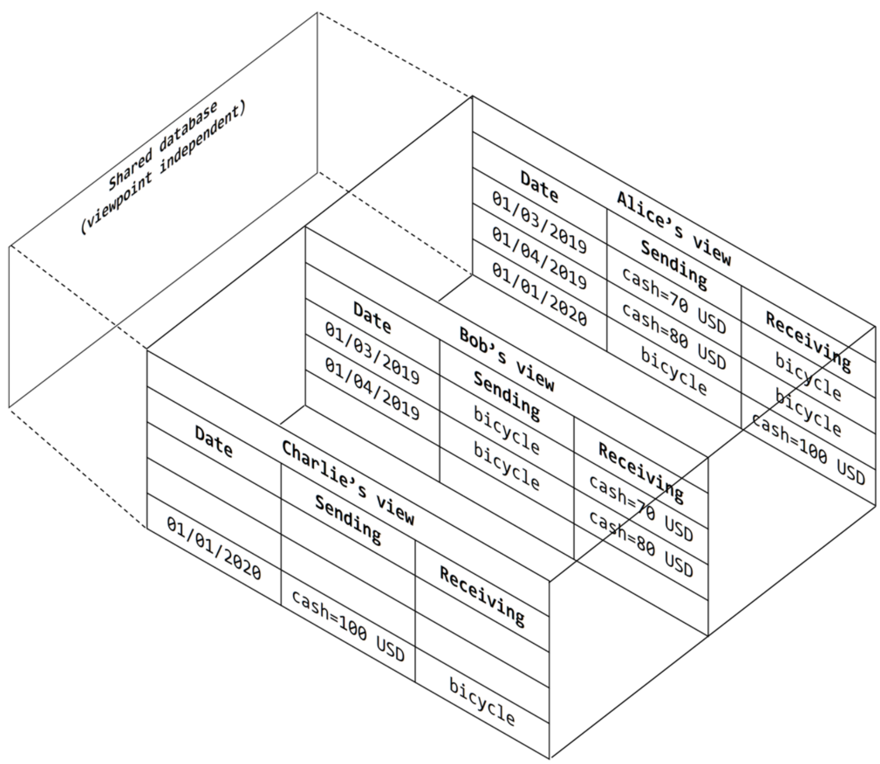

FIGURE 2: A SIMPLIFIED THREE-DIMENSIONAL SHARED RECORD OF TRANSACTIONS. THE SAME TRANSACTION RECORD MANIFESTS DIFFERENTLY FROM THE VIEWPOINT OF EACH USER, I.E. IN EACH SHEET. [14] **FIGURE 1 CONTAINS AN EXAMPLE OF A VIEWPOINT-INDEPENDENT RECORD. BASED ON BOYLE (2003A, 2003F).**

Historically, shared transaction records and three-dimensional accounting were technically infeasible. Therefore, each party to a transaction had to make its own duplicated, viewpoint-dependent record of the same transaction in its own, two-dimensional books. We may call this "redundant bookkeeping" (Boyle 2001a, 2000f). However, with Internet-based shared data environments becoming a reality, such redundancy can be eliminated.

FIGURE 3 presents a stylised comparison of the accounting schemes discussed. Bookkeeping redundancy can be eliminated through a shared ledger system constituting the single source of truth. As a generalised form of shared ledger systems, REA features a shared "collaboration space" where the economic events of agents are recorded (McCarthy 2016), but it does not specify the practical procedure necessary to agree on the single record. TEA introduces a signature-gathering process involving the two parties, plus a trusted third party. In this context, TEA can be seen as a more concrete implementation, which takes this extra step at the cost of losing generalizability (Grigg 2020a).

[14] Boyle (2003a, 2003f) noted that it was possible to add more dimensions to the three-dimensional accounting grid, e.g. to allow a breakdown by type, month and/or purpose. In that context, the 3D accounting cube becomes a hypercube.

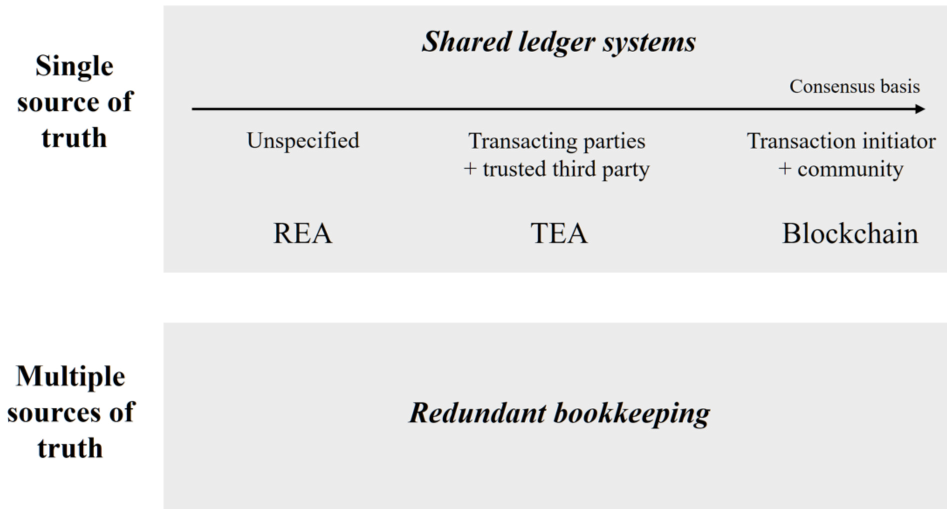

FIGURE 3: REA, TEA AND BLOCKCHAIN AS SHARED LEDGER SYSTEMS EMERGING IN CONTRAST TO REDUNDANT BOOKKEEPING. WHILE IN TRADITIONAL REDUNDANT BOOKKEEPING, MULTIPLE SOURCES THAT DESCRIBE THE SAME ECONOMIC EVENT EXIST AND VARIOUS VERSIONS CAN ARISE, SHARED LEDGER SYSTEMS INCORPORATE DESIGNS FOR A SINGLE SOURCE OF TRUTH. THE CONSENSUS BASIS ON WHICH THE TRUTH IS DETERMINED DEPENDS ON THE SPECIFIC SHARED LEDGER SYSTEM.

Blockchain technology develops the TEA model even further: it is a form of distributed software architecture that allows untrusted actors to securely agree on a transaction record without a central point of supervision (Tasca and Tessone 2019; see also Liu et al. 2019). To do so, it replaces the trusted third party featured in the original TEA design with community-based consensus. In this manner, blockchain shows that the collaboration spaces envisioned in REA are practicable with current computational possibilities (McCarthy 2016).

Accounting and Bookkeeping

As implied by its name, TEA is an *accounting* model, but its "triple-entry" component refers to a specific technical innovation at the *bookkeeping* level (signed messages). For this reason, Grigg (2019a) distinguishes between triple-entry bookkeeping (TEB) and triple-entry accounting (TEA). Similarly, REA is a generalised accounting model including a proposal for a bookkeeping innovation (the OeDBTR). This demands a brief discussion of the distinction between bookkeeping and accounting.

Bookkeeping is synonymous with recordkeeping, i.e. it is defined as simply keeping a sequential (chronological) record of transactions. Accounting, instead, builds on top of bookkeeping to make that information flow into the decision-making areas of a firm by means of systematising, compiling, collating, synthesising, processing, analysing and auditing. Such viewpoint appears to be the majority position in the peer-reviewed literature (Rukhiran and Netinant 2018; Vollmer 2003, 357) as well as in many instruction manuals and handbooks (Chandler 1977, 109-110; Ge 2005, 3; Ginigoada Foundation 2017, 3; Peters-Richardson 2011, 7; Wild, Shaw, and Chiappetta 2011, 4) and will be followed in the remainder. However, it is worth noting that earlier scholars make no distinction between bookkeeping and accounting (Lomax 1918, 74; Edwards 1960, 447).

This ambiguity may raise questions, as even though both double-entry *bookkeeping* and double-entry *accounting* are terms in usage, "double-entry" fundamentally refers to a trait of the *bookkeeping* system, not the accounting itself. However, it *is* legitimate to speak of double-entry *accounting,* because the specific accounting edifice is determined by the bookkeeping technique. Similarly, despite the notion that there is nothing intrinsically "triple" about the *accounting* in

TEA, the underlying bookkeeping innovation should impact the accounting practice, hence justifying the term triple-entry "accounting".

TEA is thus TEB *with an accounting solution*. In other words, a TEA system includes a shared transaction repository with a "signature – signature – signature" structure, but it is not limited to just sequentially storing transactions. Rather, it also serves to classify and interpret them, facilitating decision-making, financial analysis and forecasting, tax planning and financial reporting.

In the previous section, we introduced a series of transactions involving bicycles between Alice, Bob and Charlie. This triple-entry process is depicted in **FIGURE 1** and only concerns *bookkeeping*. Yet the question arises: what was Alice's profit from the transaction with Charlie?

Alice can resort to a number of methods to calculate the profit from each transaction, such as last-in first-out or LIFO, first-in first-out or FIFO, and average cost or AVCO (Peters-Richardson 2011, 104-107; Wild et al. 2011, 234-236). The choice between these methods will determine, for example, whether Alice made a profit of 20 USD, 30 USD or 25 USD in her transaction with Charlie. Nevertheless, this choice is a matter of accounting, not one of bookkeeping. Thus, implementing a shared transaction repository with a triple-entry model does not by itself answer this accounting question.

Either the choice between methods of assigning costs is left to Alice, Bob and Charlie, or it is pre-determined in the accounting software that they use. The STR may choose to integrate an accounting module (e.g. a General Ledger for Reporting; Boyle, 2001b)[15] to the TEB system used by Alice, Bob and Charlie. In other words, the STR provider may offer them to purchase a

[15] See Boyle (2001b).

subscription to his online Enterprise Resource Planning (ERP) system or webledger (a multi-user accounting suite) such that the TEB transaction records are automatically entered into each party's webledger. This may also allow Alice, Bob and Charlie to publish financial reports, or to be audited, also in real-time. This accounting module built on top of the TEB record is called triple-entry *accounting.*

Blockchain technology is also an innovation at the bookkeeping level. However, blockchain systems allow for the execution of programs that can perform accounting functions. The most important kind of program used in blockchains is the *smart contract*. Smart contracts are digital programs that automate tasks related to contract execution, documentation or control minimising the need for trusted intermediaries (Szabo 1994, 1996). These tasks may include *bookkeeping,* but also *accounting*. However, as the Bitcoin protocol only features very limited smart contract capabilities, Bitcoin may be regarded as a triple-entry *bookkeeping* system that cannot function as an accounting suite. Accounting must therefore be done almost entirely off-chain. Nevertheless, other networks may be able to perform this role to a greater extent.

Earlier in this section, we established that a number of methods can be used to calculate the profit from each transaction (e.g. LIFO, FIFO, AVCO). Furthermore, we explain that the choice between methods can be left to the parties or be pre-determined in the accounting software that they use. A blockchain supporting more complex smart contract technology, such as Ethereum or Hyperledger's Fabric, may do precisely this.

A blockchain-based system that contains a profit-calculation protocol would look as follows. For our previous example, where Alice purchased a bicycle for 70 USD and then another one for 80 USD, the price values of two units can be stored in a smart contract. Alice then sold one of the bicycles for 100 USD and would like to calculate the profit with either the AVCO, FIFO or LIFO method.

Minimum working examples of smart contracts, written in Solidity, are included for each one of these methods in *Appendix A: Profit-calculating smart contracts*. If Alice chose the AVCO method, the smart contract would calculate the difference between the value of the sale ("itemPrice") and the average cost of each element in the inventory (25 USD profit). If she chose the FIFO method, the function would subtract the cost of the first object in the array (and remove it for future calculations, so that the next object proceeds to become the new first object) from the value of the sale (20 USD profit). Finally, if she chose the LIFO method, the function would do the same as the FIFO function, but for the last object in the array (30 USD profit).

This illustrates how, although a shared ledger system buttressed by blockchain is a bookkeeping mechanism, it can be designed to support *accounting* features. Ultimately, this puts into question the distinction between accounting and bookkeeping of the previous section. While it appears intuitive that "mere" bookkeeping is different from the intellectual process of *accounting*, the line between the two becomes blurred when this intellectual task is used to create a pre-determined set of bookkeeping steps. If a task typically associated with accounting is programmed into a webledger to be performed automatically if the user fulfils certain bookkeeping steps, it is unclear (and possibly also inconsequential) whether it then constitutes either accounting or bookkeeping. For this reason, the interchangeable usage of the terms bookkeeping and accounting is plausible, as is the usage of both the terms double-entry accounting and double-entry bookkeeping, as well as TEA and TEB.

III. A Genealogy of Shared Ledger Systems

As we described in the Introduction, the *vox populi* history of shared ledger systems contains gaps, such that their very development is improperly understood.

The popular version of the story goes as follows: Pacioli invented double-entry accounting in 1494. The convention remained unchallenged until 1982, when Yuji Ijiri ideated his version of TEA (Vijai et al. 2019). However, Ijiri's ideas were forgotten until Ian Grigg brought them back to life in 2005, making a series of twists to the concept (ibid; Fullana and Ruiz 2020). In spite of this cryptographic innovation, Grigg's idea was impracticable at the time because it was necessary to *trust* a third party with the shared ledger. Yet thanks to the exogenous appearance of Satoshi Nakamoto's Bitcoin whitepaper in 2008, suddenly it was possible to implement TEA and other shared ledger systems designs without what impeded their viability: the need for trust (Cai 2019; Rao 2020).

However, this story suffers from multiple omissions and inaccuracies. Notably, it overlooks the role of Todd Boyle in authoring the concept of triple-entry accounting. In consequence, the impact of the ideas of William E. McCarthy and Robert Haugen in Boyle is also overlooked (without prejudice to the originality of Boyle's work). Hence, the influence of the REA model in the genesis of TEA is neglected.

Furthermore, while Grigg's work on this topic was first documented in 2005, much of it was undertaken between 1995 and 1997. Moreover, Ijiri's momentum accounting bore almost no relationship with Grigg's TEA. While a number of authors do point this out (Cai 2019; Dai and Vasarhelyi 2017; Gröblacher and Mizdraković 2019; Pacio 2018a; Wang and Kogan 2018), others appear to be unaware of this (Faccia and Mosco 2019; Faccia and Mostenau 2019; Vijai et al.

2019; Jeffries 2020). Note that Ijiri's exposition of accounting concepts did have a minor influence on McCarthy. Although this relationship can be interpreted as a historical connection between Ijiri and Grigg, which is not recognised in the TEA literature reviewed, it is at best an insignificant one.

Finally, blockchain was likely not introduced as a completely exogenous invention that eventually enabled TEA. Instead, the Bitcoin blockchain bookkeeping model may have been influenced *by* TEA: Boyle and Grigg's TEA were among the many ideas discussed throughout the 1990s that influenced Bitcoin.

FIGURE 4 precises the extended and corrected genealogy of shared ledger systems.

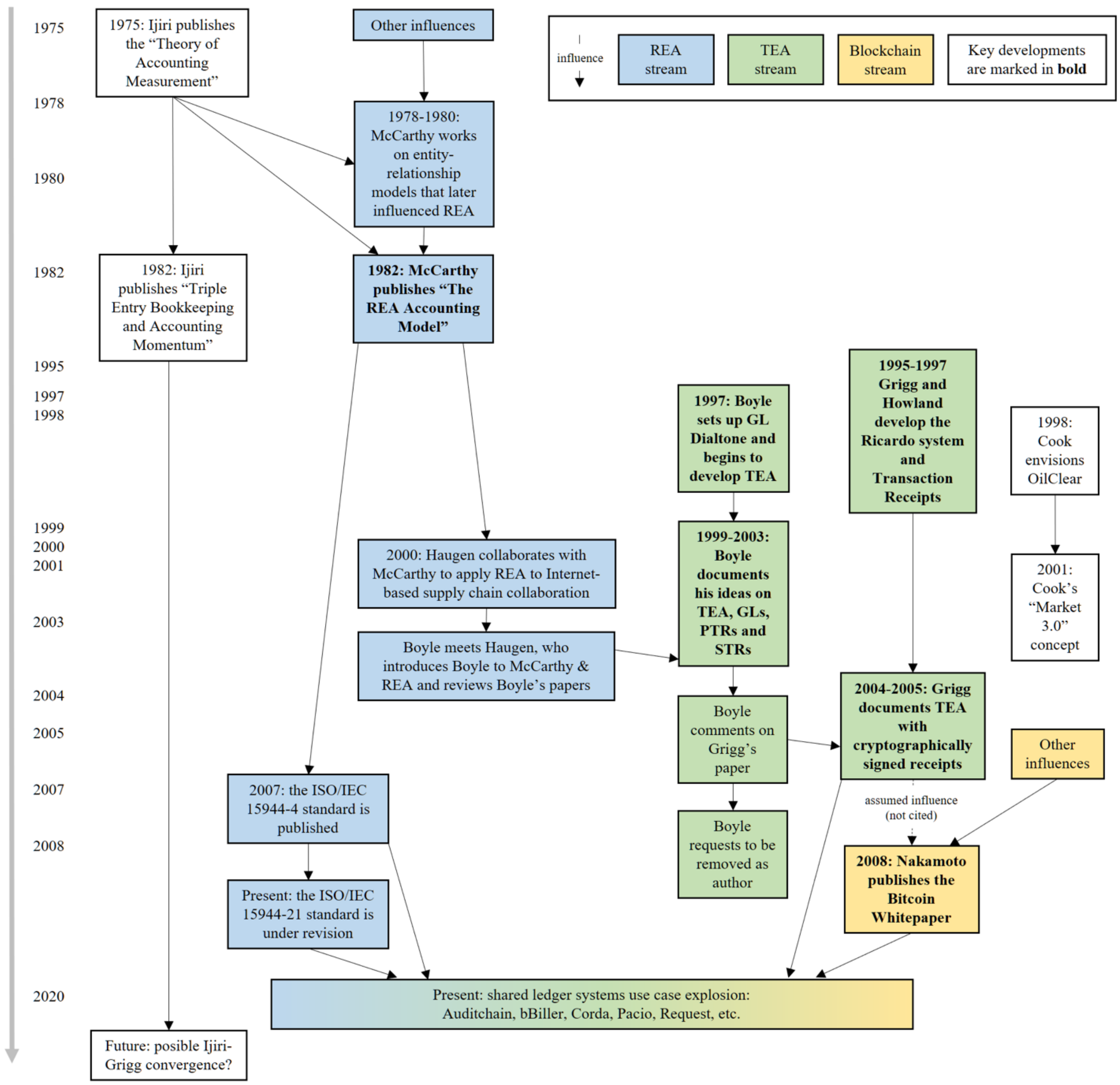

FIGURE 4: PARALLEL DEVELOPMENT AND CONFLUENCE OF THREE STREAMS OF RESEARCH ON SHARED LEDGER SYSTEMS, NAMELY REA, TEA AND BLOCKCHAIN.

This section expands on the previous corrections,[16] so as to "set the record straight" and adequately conduct a genealogy of shared ledger systems.

Early Antecedents to Triple-Entry Accounting and other Shared Ledger Systems

The year 1982 brought forth the first major innovative challenges to the accounting status quo. Accounting professors Yuji Ijiri and William E. McCarthy produced a series of groundbreaking papers.

Ijiri (1982, 1986, 1989, see also Hsieh 2018) explained that, in single-entry bookkeeping, only wealth (assets and liabilities) was recorded. The double-entry system incorporated income (flow accounts, revenues and expenses) so that, generally, one year's income statements explained the difference between two consecutive years' wealth statements:[17] the rate of change in wealth or "momentum"[18] (Ijiri 1986, 747). Thus, a third entry to explain the rate of change in *income* would constitute a logical extension that would make accounting systems "more dynamic and not focused on the present state (Balance Sheet), but on the future forecast" (Gröblacher and Mizdraković 2019, 60).

In other words, Ijiri had envisioned that a third entry would be used to explain the change between the income statements of two consecutive years, i.e. the rate of change of income or "force." Force,

[16] Also note that, while Pacioli (1494) did popularize double-entry accounting, Benedetto Cotrugli (1573) and Marino de Raphaeli (1475) had preceded him in beginning to introduce and develop the concept (Postma and Helm 2000; Sangster 2015; Sangster and Rossi 2018). Furthermore, comparable double-entry systems had been developed separately by the Italians, Koreans, and the second Muslim Caliphate at different times for the same purpose (Byeongju 2018; El-Halaby and Hussainey 2016; Zaid 2004).

[17] Setting aside changes in equity for the sake of simplicity (Ijiri 1986, 747).

[18] Which becomes "income" when multiplied by duration (Ijiri 1986, 747).

which could also be described as the *rate of change in the rate of change in wealth*,[19] is recorded in a third column named "trebit" (with debit ≡ credit ≡ trebit) together with wealth and momentum. In contrast, momentum and income are recorded in the "credit" column and assets are recorded in the "debit" column (Ijiri 1986, 751). Ijiri named this system "triple-entry bookkeeping," though it is also known as "momentum accounting."[20]

Almost simultaneously, McCarthy extended his earlier work (McCarthy 1979, 1980, 1981) on entity-relationship modelling (Dunn, Gerard, and Grabski 2016), and proposed an accounting framework for a shared data environment (McCarthy 1982, 554). McCarthy observed that, when information on the economic events of the firm did not fit the categories used by accountants, they either left it out or forcefully compelled it to fit those categories (McCarthy 1980). Due to the filtering of data, the basing of all records in the double-entry method, the aggregation of information "over time and sections" (McCarthy 1980, 512), and the application of inappropriate classification schemes, accounting data resulted in little value to other departments of the firm.[21] As a result, other departments had to develop their own information systems, which impaired integration within the firm.

In order to address these problems, McCarthy (1982) argued for a *centrally* defined database able to support multiple "views" (multiple users) of itself. The system would not be based on debits, credits, and accounts, which McCarthy (1982, 560) regarded as mere "mechanisms for manually storing and transmitting data" that reduced the usefulness of the information. For this reason, McCarthy's framework was at the antipodes of Ijiri's project to extend double-entry accounting

[19] And becomes "impulse" when multiplied by duration (Ijiri 1986, 747-748a).

[20] Ijiri also advocated cryptosystem solutions involving public encryption keys and private decryption keys to protect business confidentiality (Ijiri and Kelly 1980, 118-120).

[21] A more recent update to this criticism also included a critique for the lack of automation (McCarthy 2016).

principles. Nonetheless, McCarthy did draw some ideas from Ijiri's previous work on accounting measurements (see Ijiri 1975).[22] However, momentum accounting was not among them.

In order to build this shared data environment, McCarthy proposed recording detailed and atomistic transaction histories. This record would be underpinned by McCarthy's particular conceptual schema or "semantic model," whose advantage would lie in its "semantic expressiveness:" the high degree at which it mapped corporate reality. The semantic model consisted in entity-relationship schemata representing economic events carried out by economic agents with economic resources, and the duality relationships connecting them. As a result, REA accounting would achieve granularity, be more efficient and accurate, and identify the agents involved, as well as other details, while preserving the duality (causal relationship) of economic events (McCarthy 1982; Dunn, Gerard, and Grabski 2016). Notably, REA would enable integration: a shared transaction record for the different departments of the firm.

Other precedents to shared ledger systems should also be mentioned. Much of the work that came out from the blockchain community in the last years is related to the cypherpunk movement of the late 20th century. Concepts discussed in mailing lists back then may have had an impact in the following decades. In 1998, Nick Szabo designed BitGold and Wei Dai proposed b-money (Szabo 2008, Dai 1998), which possibly influenced Bitcoin later, together with other pre-existing components such as public-key cryptography developed in the 1970s, the hash tree (1979), cryptographic timestamps (1991), the decentralised peer-to-peer networks of the 2000s (Mainelli

[22] While McCarthy (1982, 556) states that "the REA framework (…) is explained using the ideas of a number of accounting theorists, principally Yuji Ijiri," quotes concepts, uses ideas and expresses accounting principles following Ijiri (1975) on several occasions, Holman (personal communication, February 21, 2020) makes clear that "only a few vocabulary terms were adopted in the interest of consistent terminology, and nothing more." Ijiri (1993) defended the "beauty of double-entry bookkeeping," which McCarthy radically opposed. Thus, connections between the two authors can only be properly drawn if adequately contextualized, so as not to present them as allied researchers.

and Smith 2015), Byzantine fault tolerance (Swanson 2015) and the Proof of Work mechanism developed by Adam Back (2002) in Hashcash.

Szabo's influence over Bitcoin is unclear, as Szabo's BitGold was not cited in the Bitcoin whitepaper. While Dai's b-money was cited, it was only referenced as a source for the statement that, absent a trusted third party, transactions must be publicly announced to avoid double-spending. According to Wei Dai (2011) himself, Nakamoto had not even read his work until after having re-invented the idea independently.

Mainelli and Smith (2015; see also Swanson 2015) identify other early examples of distributed ledgers with features in common with blockchain technology, including WebDNA (1995), Z/Yen's semi-distributed encrypted ledger (1996), Stanford University's "Controlled Lots of Copies Keep Stuff Safe" (CLOCKSS) and "Lots of Copies Keep Stuff Safe" (LOCKSS) (1999), and Ryan Fugger's Ripple (2004; see also Nakamoto 2009). There is, however, no documented impact of these concepts on the Bitcoin whitepaper.

Finally, Eric Hughes' "open book accounting" method also deserves mention. Back in 1993, Hughes (1993a) proposed a public transaction record with encrypted private balances. Such a register would be a "single entry account" in a "shared funds account" that can moreover be expressed by double-entry bookkeeping for the parties. The accounts would be kept in accordance with each other through a public verification method (Hughes 1993b). Hughes, however, could not make this idea work technically (Grigg 2018).

The Emergence of Shared Ledger Systems

> "We're followers of McCarthy's economic ontology, and ISO 15944-4." –
>
> Todd Boyle (2003c).

The 1990s and early 2000s brought about many important developments for shared ledger systems. Todd Boyle designed webledgers and coined the term TEA, Ian Grigg and Gary Howland devised the Ricardo Payment System, Chris Cook developed OilClear, McCarthy and Holman extended REA to multiple-company accounting, and REA-based ISO/IEC standards began to be developed.

GL Dialtone and Webledgers

In 1997, Todd Boyle (2001d, 2001e) moved back from Japan and set up General Ledger Dialtone in Seattle, an accounting solutions company specialised in webledgers. While Boyle had *independently* come up with the idea of shared ledgers, he was later influenced by one of McCarthy's collaborators, Robert Haugen. Haugen was a software developer for a Core Components ebXML standards team (Boyle 2015 in Grigg 2014), who had worked in applying McCarthy's REA to supply chain Internet-based collaboration (Haugen and McCarthy 2000). Haugen introduced Boyle to McCarthy's work first, and McCarthy himself later, which had an impact on Boyle's ideas, consolidating his proposal for a shared ledger system.

Boyle (2000c) believed that McCarthy's REA framework was "high level (...) ahead of its time (...) [and] a goldmine" with many merits,[23] but envisioned another application for it: TEA. This concept would be commercially launched as an Internet-based, multi-company, and low-cost accounting software. It would enable large supply chains (value networks) supported through a webledger "spitting out" the transactions therein (ibid), i.e. a "general ledger (...) where independent companies could post their resource transfers" (Boyle 2015 in Grigg 2014; see also Haugen and McCarthy 2000). The webledger could contain all internal general ledger entries but its principal use was to "contain those journal rows involving external parties" (Boyle 2001g). To enable automatic reconciliation and external (as well as internal) integration, inter-ledger semantics based on the GAAP would have been developed to allow different business systems to interface and form a coherent whole (Boyle 2002).

This was designed to have a classic double-entry accounting structure but a non-double-entry interface, for user friendliness (Boyle 2000a), which reflects another of Boyle's criticisms of REA: he thought that, even though REA was superior to double-entry, the quest to replace the latter with the former was "a distraction," as double-entry "merely" records data alongside a business system (Boyle 2000c; see also Boyle 2000a, McCarthy 2001).

For this purpose, as well as for other commercial reasons, Boyle believed that a mechanism to communicate between both parties in a transaction where and when an economic exchange has happened ("recognition") should be built in a joint web accounting application (a B2B middleware server; Sachs 2001 in Boyle 2001c; Boyle 2001g), rather than delegated to each participant (Boyle

[23] Boyle also criticized REA for being mislabeled as an accounting system (when, in his view, it was really a very generalized business information system). He moreover believed that the circumstances that REA had originally come to solve in the 1980s did not exist anymore (Boyle 2000c). For a response, see Haugen (2001) and McCarthy (2001).

2001c). The mechanism should act as an encrypted "public document repository service," as a "notification service," as a service to record replies (e.g. acceptances), and as an archive and reporting service to provide "persistent and responsive storage of inter-party transactions, sufficient to achieve a robust and intrinsic reconciliation" (Boyle 2001g).

Boyle's webledger architectures would implement this solution in the form of a "shared" or "public transaction repository" (STR or PTR) based on "single-entry hosted transaction tables" (Boyle 2000e, 2000d, 2001a). There would thus be a single, shared, network-centric record, but, because of the two additional private stubs, the system would be called *triple*-entry accounting (Boyle, 2001b). Boyle (2000c) proposed REA to support the model for the back-end software and even developed a REA-based economic ontology to describe this system conceptually (Boyle 2003d).

> "Thanks to Bill McCarthy and his REA school, who were the source of most of these ideas" – Todd Boyle (2003d).

OilClear and Market 3.0

At approximately the same time that Boyle developed GL Dialtone and TEA, oil markets consultant and researcher Chris Cook independently developed OilClear, a petroleum-specific STR concept. Cook (2002) argued that "a 'shared transaction repository' and a 'shared title repository' (...) connected by clearing and settlement software" were necessary for the new market structure in the age of the Internet and instantaneous communication. The concept, called "Market 3.0," reportedly hit the "Internet neutrality-liquidity paradox," did not find a route to the market and was later fragmented and appropriated by ICE's eConfirm and by CME's Tradehub. While

ICE's eConfirm and CME's Tradehub did not stop resorting to exchanges (Cook 2016), OilClear set a milestone nonetheless.

The Ricardo Payment System

Also independently from Boyle and Cook (Grigg 2019b), Ian Grigg together with Gary Howland co-developed a similar concept between 1995 and 1997: the Ricardo payment system and Ricardian contracts, documented in Howland (1996).[24] This idea, which Grigg (2000, 2004, 2005) kept developing afterwards, was an attempt to replicate how economic events are recorded internally *within* firms through an ERP system, in a shared data environment between firms. It involved a shared set of receipts for the transactions two parties have in common, and a trusted third party limited to signing, timestamping and ordering. While, originally, two different receipts were conceived: one for the payer and one for the payee (Howland 1996), shortly after this idea was abandoned, folding the two receipts into a single one. This adjustment constituted the genesis of triple-entry.[25]

This shared receipt would be the *dominating* record for a transaction. Moreover, in Grigg's design, the receipt is not just evidence for the transaction: it *is* the transaction itself because it holds all the relevant information to build an entire data processing concept around it.[26] Furthermore, to prevent any disputes around semantics, these are locked down through a Ricardian Contract (Grigg 2005b): a human and machine-readable text file containing both the terms of an agreement and the program executing the financial instrument, such that they are the same thing, i.e. "the issue is a contract"

[24] Ian Grigg was not explicitly credited in this paper. However, the paper states that the company founded by Grigg (Systemics Ltd.) developed the system, and Grigg himself reports being a co-developer in Grigg (2000).
[25] Grigg, I. (personal communication, April 15, 2020).
[26] For example, it is possible to calculate Bob's balance by reading and calculating the net sum of all the receipts mentioning Bob.

(Grigg 2004, 2000). The parties hold a (cryptographic) key to authorise each transaction and a copy of the receipts issued by the accounting agent (see also Boyle 2003b). To modify the record, the accounting agent needs the signature of both parties. In other words, every modification of the record requires a three-party consensus.

The Convergence of Shared Ledger Systems: The Forgotten Influence of Todd Boyle and William McCarthy

Confluence of Streams

In 2000, Grigg (2000) began to document his work. In 2004, upon realising that his design could have radical implications for accounting, Grigg pursued further development of his ideas (ibid), which he called triple-entry accounting.[27] Grigg's reasons for this name were different from Boyle's: instead of the "stub – shared entry – stub" structure, the system owed its name to the "signature – signature – signature" (or Boyle's "offer – acceptance – validation") structure.

A draft of the resulting paper was circulated in June 2005, with Boyle commenting on it. Boyle noted that he had been working on the same idea for years as well (Grigg 2016a). As a consequence, Grigg integrated and implemented many of Boyle's ideas within the paper. However, while a draft of the paper "credited Todd Boyle as an author, (...) this was later withdrawn at his request due to wider differences between the views" (Grigg 2005b). These differences were related to the breadth of the scope or generalizability of the model.

[27] In more recent years, Grigg (2019a) has argued that his 2004 paper should have been titled "triple-entry *bookkeeping*" instead of "triple-entry *accounting*," since he had addressed the reliability of the record only.

Grigg first became aware of McCarthy's REA concept in 2017 through Boyle introduced (Grigg 2017a), but McCarthy's influence was present nonetheless. He was also unaware of Ijiri's work (Grigg 2020), though the latter's imprint on TEA had certainly been almost trivial.[28] Note that, according to Grigg (2017a, 2017b, 2017c, 2020a), his TEA inadvertently implemented key ideas more generally contained in REA: "the Receipt as I describe it in the paper and as it is used, is an REA construct converted to data; the (hash of the) Ricardian contract is the resource, the signing/timestamping by [the STR] is the event, and the payee/payer are the agents."[29]

Blockchain and TEA: A Two-way Street?

However, before the advent of blockchain, a workable triple entry system would have necessitated a trusted third-party intermediary, who would also have been susceptible to attack, error, or loss. The intermediary would have been vulnerable, like the transacting parties themselves. The invention of blockchain technology permitted an adaption of Grigg's theory without a single center (Grigg 2019c).

This development was brought about by Satoshi Nakamoto's (2008) Bitcoin. The new solution renders moot the central intermediary with a decentralised ledger to which each party's books are connected, in which both sides of a transaction are recorded, and thus having the entries reach a

[28] Ijiri's triple-entry bookkeeping is not the only namesake to the system ideated by Ian Grigg and Todd Boyle. In the nineteenth century, Russian theorist Fedor Venediktovich Ezersky also designed a system named "triple-entry accounting", also known as "Russian triple-entry" or "triple book" system (Faccia et al. 2020). Ezersky proposed a system characterized by using only three books (capital book, systematic accounts book and balance book) and by a continuous update of inventory (instead of at the end of periods) to avoid lags (Victorovna 2015).
[29] Grigg, I. (personal communication, April 15, 2020). For further discussion on how TEA may offer an implementation of more generalized ideas contained in REA, see Grigg (2020b).

consensus. This application may be Business-to-Business (B2B) or Government-to-Government (G2G), e.g. between companies' tax and royalty payments to governments.

As stated before, Bitcoin had been influenced by public-key cryptography, peer-to-peer technology and, possibly, BitGold and b-money. However, there are reasons to suggest that TEA was also one of the influences over blockchain, instead of only being enabled by it. Firstly, there are remarkable architectural similarities between Boyle and Grigg's ideas and Bitcoin: Bitcoin is a pseudonymous, immutable public transaction repository with an integrated payment layer underpropped by a triple-entry structure (in which the trusted third party is the distributed ledger or community).

Moreover, there are anecdotal reasons to think that TEA could have been one of many sources in the corpus of preceding work on top of which the Bitcoin edifice was built. In discussions within the cryptographic community, Grigg and Boyle's work have been regarded as direct influences over blockchain (Brown and Grigg in Brown 2015; Cook 2013, Grigg 2011, Grigg 2014, Grigg in Swanson 2015; Sleeter 2014; Smith 2019, Wright 2019). In addition, Boyle and Grigg were part of the same cypherpunk mailing lists in which cited influences in the Bitcoin whitepaper like Adam Back and Wei Dai participated (see Nakamoto 2008; Venona Cypherpunks Archives 2004).[30] Nonetheless, since Nakamoto's identity has not been confirmed (and might never be), this evidence is limited.

[30] Some believe Grigg *is* Satoshi Nakamoto himself, based on, inter alia, stylometric studies (Helsel 2018, Smith 2019), a claim we are unable to verify. Grigg (2016c) has denied being a member of the Satoshi Nakamoto team. However, Grigg (2016b) has also claimed direct knowledge of its internal workings.

The Relevance of Ongoing Work in REA to TEA

As stated above, Grigg was unaware of McCarthy's REA model and his influence over Boyle's work. Considering also that most TEA use cases and papers follow Grigg's idea (not Boyle's), it is therefore unsurprising that recent developments in the REA world have remained unnoticed despite their high pertinence to TEA. Nevertheless, the developments continue, with the work of ISO/IEC JTC 1/SC 32/WG 1 – a working group of a subcommittee of the joint committee between ISO and IEC – being most relevant.

The ISO/IEC 15944-4 standard was published in 2007,[31] then updated in 2015 and is currently under review. It uses the REA ontology to model a formal framework for business transactions named "Open-edi Business Transaction Ontology (OeBTO)" (ISO/IEC 2015, v; see also Dunn et al. 2016, 555). This framework maintains that the redundancy in mirroring records of a transaction must be abandoned to eliminate the possibility of inconsistencies (ibid, vi) and because it is viewpoint-dependent. In turn, it proposes an independent, inter-enterprise view of transactions.

While ISO/IEC 15944-4 mostly provides definitions, the joint work of Holman and McCarthy (2019, see also Holman 2019) built upon it in designing ISO/IEC 15944-21, a standard providing guidance on the implementation of an OeDBTR, i.e. a shared transaction repository (typically, but not necessarily, blockchained) remarkably similar to a TEA system, within the REA ontology. A draft of this standard has already been registered and approved, but the publication process has yet to be completed (ISO 2019).

[31] Boyle (2003c) claimed to be a follower of ISO 15944-4 in 2003, which means that he was aware of the draft before publication.

This development has interesting implications, as it opens the door for TEA systems to follow ISO/IEC specifications. In fact, one of the TEA use cases listed in **FIGURE 4**, bBiller, is an OeDBTR implementing the REA ontology. Another TEA use case, Pacio, incorporates the REA ontology in the Standardised Semantic Information Model Database of Facts in the TEA and IDEA diagrams of its whitepaper (Pacio 2020, 6, 9). This facilitates the possibility of a TEA-REA reconciliation, in spite of the neglected influence of REA in TEA.

Moreover, further convergence is conceivable. We established that Yuji Ijiri did not influence TEA, except for the adoption by McCarthy of the terminology laid out in Ijiri's works prior to momentum accounting. Nevertheless, Ian Grigg (2020b) has recently stated that there is a potential application of TEA for Ijiri's *momentum accounting* and: Ijiri's momentum accounting requires complicated calculations that rely on the absolute accuracy of the underlying records, making the model too idealistic to be feasible for the market (ibid). The execution of momentum accounting on top of cryptographic triple-signed receipts, however, might allow the model to perform reliably.

IV. DISCUSSION

Shared ledger systems constitute an unprecedented innovation milestone. They are not a panacea in that they do not replace many of the traditional functions of accounting and do not automatically prevent fraud, money laundering, etc., by themselves. Nevertheless, shared ledger systems have led the way for more efficient and transparent accounting applications in the Internet era. In particular, TEA is one of the pioneering concepts for accounting in shared data environments, delivering many benefits by enabling or facilitating external integration, instantaneous reconciliation, lower redundancy, low-cost real-time auditing, financial reporting, invoice automation, dispute resolution, etc. (Alawadhi, et al. 2015; Boyle 2002, 2003f; Dai 2017; Dai and Vasarhelyi 2017; ICAEW 2018; Mohanty 2018, 47; Request 2018a, 2018b)

The lack of an integral genealogy of TEA has obscured the role of the *accounting discipline* in giving birth to TEA. Specifically, Boyle's work was overlooked, and so was the influence of the REA model over TEA. Consequently, the point that TEA is to some extent a historical byproduct of McCarthy's research was rarely raised in public discourse. As a consequence, REA and TEA have remained two separate streams of research. Yet, it would be of great interest to bring these streams together.

Moreover, since it is conceivable that TEA was one of the many inspirations behind Satoshi Nakamoto's Bitcoin, McCarthy's REA model might have had an indirect historical impact on the genesis of the blockchain technology itself. It is often said that "blockchain is fundamentally an accounting technology" (ICAEW 2018, 1). In the light of these findings, this statement may be truer than ever thought before.

Indeed, the structural resemblances between REA and TEA are not just a coincidence but a natural outcome given the historical influence of the former over the latter. This may explain why bBiller, for instance, is considered a TEA use case (Pacio 2018b), but their designers consider it a REA use case. It may also explain why Pacio incorporates part of the REA ontology in its TEA design. Furthermore, although it remains unclear whether TEA influenced the original Bitcoin blockchain, influence would provide a natural explanation for blockchain projects exhibiting TEA features, even if without deliberately implementing the idea nor necessarily using the term.

V. CONCLUSION

In this paper, we attempt to trace three intersecting development pathways that represent incarnations of shared ledger systems. In particular, we explore possible connections among the long-established REA and TEA frameworks, and the nascent blockchain technology. By filling in the gaps in the genealogy of shared ledger systems, we correct historical misconceptions, and give due credit to related prior works that have been insufficiently recognised. A clearer understanding of the historical evolution of shared ledger systems potentiates further cross-pollination in academic and practitioner circles, in particular between proponents of Resource-Event-Agents, triple-entry accounting and blockchain.

REFERENCES

Alawadhi, A., Ames, B., Maciel de Aquino, C. E., Arthúrsdóttir, M., Brennan, G., Brown-Liburd, H. L., Bumgarner, N., Byrnes, P., Criste, T. R., Ghosh, S., Gross, J. A., Gullkvist, B., Hardy, C., Jónsson, H. M., Lasslet, G., Medinets, A., Moon, D., Miyaki, E. H., Sigolo, N., Sigurjónsson, S., Stewart, T. R., Teeter, T., Vasarhelyi, M. A., and Warren, J. D. 2015. Audit Analytics and Continuous Audit: Looking Toward the Future. *American Institute of Certified Public Accountants*. New York, United States of America: AICPA.

Auditchain. 2018. Auditchain: Decentralized Continuous Audit & Reporting Protocol Ecosystem. *Auditchain Whitepaper*. Available at https://auditchain.com/Auditchain-Whitepaper.pdf

Back, A. 2002. Hashcash - A Denial of Service Counter-Measure. *Hashcash.* Available at http://www.hashcash.org/papers/hashcash.pdf

Boyle, T. 2000a. CDEA General Ledger Schema. *Ledgerism*. Available at https://web.archive.org/web/20071001201740/http://www.ledgerism.net/GLSchema1.txt

Boyle, T. 2000b. Draft Schema for General Ledger. *Ledgerism*. Available at http://web.archive.org/web/20051003224536/http://ledgerism.net/GLSchemaDraft1.txt

Boyle, T. 2000c. Re: Company centric view ends. Data goes to the Net. Available at http://linas.org/mirrors/www.gldialtone.com/2001.07.14/ToddREAPost.htm

Boyle, T. 2000d. The Public Transaction Repository (PTR) project. *GL Dialtone*. Available at http://web.archive.org/web/20000622133330/http://www.gldialtone.com/PTR.htm

Boyle, T. 2000e "Webledger." *GL Dialtone*. Available at http://web.archive.org/web/20000610172851/GLDialtone.com/webledger.htm

Boyle, T. 2000f. WebLedger Architectures; Ending Data Duplication. *GL Dialtone*. Available at http://web.archive.org/web/20000829141221/http://www.gldialtone.com/endredundancy.htm

Boyle, T. 2001a. End Data Duplication. *GL Dialtone*. Available at https://linas.org/mirrors/www.gldialtone.com/2001.07.14/endredundancy.htm

Boyle, T. 2001b. GLT and GLR: component architecture for general ledgers. *GL Dialtone*. Available at https://linas.org/mirrors/www.gldialtone.com/2001.07.14/GLT-GLR.htm

Boyle, T. 2001c. RE: Initial Draft of CPP-CPA Specification. Available at http://lists.ebxml.org/archives/ebxml-bp/200101/msg00089.html

Boyle, T. 2001d. Resume. *GL Dialtone*. Available at http://web.archive.org/web/20010120224200fw_/http://www.gldialtone.com/resume.htm

Boyle, T. 2001e. Services. *GL Dialtone*. Available at http://web.archive.org/web/20010120224200fw_/http://www.gldialtone.com/services.htm

Boyle, T. 2001f. The Shared Transaction Repository (STR). Accountant's Explanation. *GL Dialtone.* Available at https://linas.org/mirrors/www.gldialtone.com/2001.07.14/STRaccountant.htm

Boyle, T. 2001g. The Shared Transaction Repository (STR) ver. 0.60 spec. *GL Dialtone*. Available at http://linas.org/mirrors/www.gldialtone.com/2001.07.14/STR.htm

Boyle, T. 2002. Inter-ledger Semantics and Methods: Requirements. *GL Dialtone*. Available at http://web.archive.org/web/20021128152948/http://www.gldialtone.com/

Boyle, T. 2003a. Accounting Hypercubes on the Internet. *Ledgerism*. Available at http://web.archive.org/web/20060102014117/http://www.ledgerism.net/hypercub.htm

Boyle, T. 2003b. File-Based Commerce (FBC) Architecture ver. 0.31 spec. Business Requirements Viewpoint. *Ledgerism*. Available at http://web.archive.org/web/20050211111755/http://ledgerism.net/fbc.htm

Boyle, T. 2003c. General Ledgerism. *Ledgerism*. Available at http://web.archive.org/web/20060425114848/http://ledgerism.net/

Boyle, T. 2003d. Ledger Ontology. *Ledgerism*. Available at http://web.archive.org/web/20080209023044/http://ledgerism.net:80/LedgerOntology.htm

Boyle, T. 2003e. Shared Transaction Repository (STR) ... a High Level Software Specification. *Ledgerism*. Available at http://web.archive.org/web/20060102052824/http://www.ledgerism.net/STR.htm

Boyle, T. 2003f. The Argument to Abolish Double-Entry Accounting and Abolish the Assets=Liabilities + Owner's Equity Equation. *Ledgerism*. Available at http://web.archive.org/web/20031208235246/http://ledgerism.net/GLSchema2.txt

Brown, R. G. 2015. How to Explain the Value of Replicated, Shared Ledgers from First Principles. *Richard Gendal Brown: Thoughts on the Future of Finance*. Available at https://gendal.me/2015/04/27/how-to-explain-the-value-of-replicated-shared-ledgers-from-first-principles/

Brown, R. G. 2016. H On Distributed Databases and Distributed Ledgers. *Richard Gendal Brown: Thoughts on the Future of Finance*. Available at https://gendal.me/2016/11/08/on-distributed-databases-and-distributed-ledgers/

Brown, R. G. 2020. Blockchain's 'Troughs of Disillusionment' Are Really The 'Trenches of Deployment.' *Forbes*. Available at https://www.forbes.com/sites/richardgendalbrown/2020/01/09/blockchains-troughs-of-disillusionment-are-really-the-trenches-of-deployment/

Byeongju, H. 2018. An Introduction on Traditional Double Entry Bookkeeping of Korea. *8th World Congress of Korean Studies*.

Cai, C. W. 2019. Triple-entry Accounting with Blockchain: How far Have we Come? *Accounting & Finance*. Accounting and Finance Association of Australia and New Zealand.

Chandler, A. D. 1977. The Visible Hand: Managerial Revolution in American Business. Harvard, United States of America: Harvard University Press.

Cook, C. 2002. Market 3.0: the Final Version. *Mondovisione*. Available at https://mondovisione.com/exchanges/handbook-articles/market-30-the-final-version/

Cook, C. 2013. Three Dimensional Accounting - Chiralkine Redux. *European Tribune.* Available at https://www.eurotrib.com/story/2013/12/29/135044/72

Cook, C. 2016. Fintech 2.0. *P2P Foundation*. Available at https://blog.p2pfoundation.net/fintech-2-0/2016/10/06

Cotrugli, B. 1573. Della Mercatura et del Mercante Perfetto. Libri Quattro. Available at https://archive.org/details/bub_gb_Ohloe05v0-oC

Dai, J. 2017. Three Essays on Audit Technology: Audit 4.0, Blockchain and Audit App. Doctoral dissertation within the Doctor of Philosophy Graduate Program of The State University of New Jersey.

Dai, J., Vasarhelyi, M. A. 2017 Toward Blockchain-Based Accounting and Assurance. *Journal of Information Systems*, 31 (3), 5-21.

Dai, W. 1998. B-money. *Wei Dai*. Available at http://www.weidai.com/bmoney.txt

Dai, W. 2011. Comment by Wei_Dai. In Making Money With Bitcoin? Article by Clippy, 2001. *Lesswrong*. Available at https://www.lesswrong.com/posts/ijr8rsyvJci2edxot/making-money-with-bitcoin?commentId=hbEu9ue9eymNzaF2J#hbEu9ue9eymNzaF2J

Dunn, D., Gerard, G. J., Grabski, S. V. 2016. Resources-Events-Agents Design Theory: A Revolutionary Approach to Enterprise System Design. *Communications for the Association for Information Systems*, 38, 554-595.

Edwards, J. D. 1960. Early Bookkeeping and its Development into Accounting. *Business History Review*, 34 (04), 446–458.

El-Halaby, S., Hussainey, K. 2016. Contributions of Early Muslim Scholars to Originality of Bookkeeping-system. *Corporate Ownership and Control*, 13 (3), 543-560.

Faccia, A., and Mosco, D. 2019. Understanding the Nature of Accounts Using Comprehensive Tools to Understand Financial Statements. *Financial Markets, Institutions and Risks*, 3 (3), 18-27.

Faccia, A., and Moştenau, N. R. 2019. Accounting and Blockchain Technology: From Double-entry to Triple-entry. *The Business and Management Review*, 10 (2), 108-116.

Faccia, A., Moşteanu, N. R., and Leonardo, L. P. 2020. Blockchain Hash, the Missing Axis of the Accounts to Settle the Triple Entry Bookkeeping System. *ICIME 2020: Proceedings of the 2020 12th International Conference on Information Management and Engineering*, September 2020, 18-23.

February, S. M. 2020a. Grokking Bitcoin. *TwoStack*. Available at https://www.twostack.org/docs/working-with-bitcoin/what-is-bitcoin/what-is-bitcoin/

February, S. M. 2020b. Triple Entry Accounting. *TwoStack*. Available at https://www.twostack.org/docs/working-with-bitcoin/transactions/triple-entry-accounting/triple-entry-accounting/

Fullana, O., and Ruiz, J. 2020. Accounting Information Systems in the Blockchain Era. *Social Science Research Network (SSRN)*. Available at https://ssrn.com/abstract=3517142

Gal, G., McCarthy, W. E. 1986. Operation of a Relational Accounting System. *Advances in Accounting*, 3, 83-112.

Ge, J. 2005. Accounting English. Available at http://abook.cn/pdf/H-0535.0101.pdf

Gee, J., and Button, M. 2019. The Financial Cost of Fraud 2019: The Latest Data from Around the World. Crowe Clark Whitehill.

Geerts, G. L., and McCarthy, W. E. 2006. Policy-Level Specifications in REA Enterprise Information Systems. *Journal of Information Systems*, 20 (2), 37-63.

Ginigoada Foundation. 2017. Introduction to Bookkeeping: A Flexible Learning Course. Available at http://oasis.col.org/bitstream/handle/11599/2749/2017_Ginigoada-Bisnis-Foundation_Introduction-to-Bookkeeping_.pdf

Goldberg, L. 1965. An Inquiry into the Nature of Accounting. American Accounting Association.

Gomaa, A., Gomaa, M., and Farag, M. 2019. The Creation of One Truth: Single-Ledger Entries for Multiple Stakeholders Using Blockchain Technology to Address the Reconciliation Problem. Social Science Research Network (SSRN). Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3397684

Grigg, I. 2000. Financial Cryptography in 7 Layers. *Systemics Ltd.* Available at https://iang.org/papers/fc7.html

Grigg, I. 2004. The Ricardian Contract. *Systemics Inc.* Available at https://iang.org/papers/ricardian_contract.html

Grigg, I. 2005a. Ian Grigg – Triple Entry Accounting. *Financial Cryptography*. Available at https://www.financialcryptography.com/mt/archives/000501.html

Grigg, I. 2005b. Triple Entry Accounting. *Systemics Inc*. Available at https://iang.org/papers/triple_entry.html

Grigg, I. 2011. Is BitCoin a triple entry system? *Financial Cryptography*. Available at https://financialcryptography.com/mt/archives/001325.html

Grigg, I. 2014. Who Invented the Shared Repository Idea: Bitcoin, Boyle and History. *Financial Cryptography*. Available at http://financialcryptography.com/mt/archives/001469.html

Grigg, I. 2016a. In a Nutshell: Ian Grigg's Ricardian Contracts and Digital Assets Prehistory (Crain, B. F.; Roy, M., interviewers). *Bits on Blocks: Thoughts on Blockchain*

Technology. Available at https://bitsonblocks.net/2016/11/22/in-a-nutshell-ian-griggs-ricardian-contracts-and-digital-assets-prehistory/

Grigg, I. 2016b. Satoshi is Dead - Long Live Satoshi - Team Leader Comes Out. *Financial Cryptography*. Available at http://financialcryptography.com/mt/archives/001593.html

Grigg, I. 2016c. @tomerkantor Nope - I am not a member of the team. *Twitter Post*. Available at https://twitter.com/iang_fc/status/727068366129655810

Grigg, I. 2017a. My thanks to Todd Boyle, originator of the concept of triple entry, for pointing out this video by Bill McCarthy https://t.co/rwGAZZaHfo. *Twitter Post*. Available at https://twitter.com/iang_fc/status/939258799654924288

Grigg, I. 2017b. "But it's not modelled there, it's only modelled up in the cloud." HOLYSMOKINMUDDERODUKS! That's triple entry accounting at the triple A. *Twitter Post.* Available at https://twitter.com/iang_fc/status/939243253060329475

Grigg, I. 2017c. Wildest dream: REA (triple entry accounting) is the Golden Spike that brings the conventional industry together with wild-eyed blockchain… *Twitter Post*. Available at https://twitter.com/iang_fc/status/939256814314622976

Grigg, I. 2018. Independently, @cjenscook invented something to STR similar for oil trading. Also someone (Eric Hughes?) postulated the "we all share a common ledger" idea on cypherpunks, but couldn't make it work technically. *Twitter Post*. Available at https://twitter.com/iang_fc/status/1035845304556417025

Grigg, I. 2019a. It's tricky - term was coined 20 years ago by Todd Boyle. My paper was 2004 or so. Oddly - paper is "3E accounting" but techniques are really bookkeeping. Yuri Ijiri's work was originally labelled "triple entry bookkeeping" but is really accounting… Confusion abounds… *Twitter Post*. Available at https://twitter.com/iang_fc/status/1182660123245928449

Grigg, I. 2019b "Triple Entry Accounting was theorised originally by Todd Boyle, who wrote of its application & implementation in the 1990s. My team implemented it independently. In 2004 I wrote the paper, thinking we'd discovered it de novo. An example of science moving in parallel, in mass." *Twitter Post*. Available at https://twitter.com/iang_fc/status/1103047183287418885

Grigg, I. 2019c. Triple entry accounting was about creating facts. Nominally for accounting, also useful for cash. Bitcoin showed a way to do 3E without a single center. Corda showed a way to do 3E with selected notaries. I'm working with 3E by communities. The future is triple entry. *Twitter Post*. Available at https://twitter.com/iang_fc/status/1103046077954097158

Grigg, I. 2020a. REA - Some Thoughts on Relationship to TEA and Computer Science. *Financial Cryptography*. Available at http://financialcryptography.com/mt/archives/001661.html

Grigg, I. 2020b. Thoughts on Momentum Accounting. *Financial Cryptography*. Available at http://financialcryptography.com/mt/archives/001655.html#more

Gröblacher, M., and Mizdraković, V. 2019. Triple-entry Bookkeeping: History and Benefits of the Concept. *Digitisation and Smart Financial Reporting*, 58-61

Haugen, R., and McCarthy, W. E. 2000. REA, a semantic model for Internet supply chain collaboration. *OOPSLA Workshop on Business Object Components: Enterprise Application Integration*, OOPSLA.

Haugen, R. 2001. Re: Company Centric View Ends. Data Goes to the Net. Available at http://linas.org/mirrors/www.gldialtone.com/2001.07.14/BobHaugenPost.htm

Helsel, B. 2018. The Uncovering of Satoshi Nakamoto Through Literary Habits? *CryptoCoin Mastery*. Available at https://cryptocoinmastery.com/the-uncovering-of-satoshi-through-literary-habits/

Holman, G. K. 2019. Blockchain and eBusiness. *LinkedIn*. Available at https://www.linkedin.com/pulse/blockchain-ebusiness-ken-holman/

Howland, G. 1996. Development of an Open and Flexible Payment System. *Systemics Ltd*. Available at http://www.systemics.com/docs/sox/overview.html (19/04/2020)

Hughes, E. 1993a. PROTOCOL: Encrypted Open Books. *Cypherpunks Mailing List*. Available at https://marc.info/?l=cypherpunks&m=85281390301301&w=3

Hughes, E. 1993b. PROTOCOL: Encrypted Open Books. *Cypherpunks Mailing List*. Available at https://marc.info/?l=cypherpunks&m=85281390300988&w=3

Hsieh, P. G. 2018. The Intellectual Development of Yuji Ijiri. *Accounting History*, 23 (4), 541-554.

ICAEW. 2018. Blockchain and the Future of Accountancy. *Institute of Chartered Accountants in England and Wales*. Available at https://www.icaew.com/technical/technology/blockchain/blockchain-articles/blockchain-and-the-accounting-perspective

Ijiri, Y. 1975. Theory of Accounting Measurement. American Accounting Association: Sarasota.

Ijiri, Y. 1982. Triple-entry Bookkeeping and Income Momentum. American Accounting Association: Sarasota.

Ijiri, Y. 1986. A Framework for Triple-Entry Bookkeeping. *The Accounting Review*, 61 (4), 745-759.

Ijiri, Y. 1989. Momentum Accounting and Triple-entry Bookkeeping: Exploring the Dynamic Structure of Accounting Measurements. Sarasota, United States of America: American Accounting Association.

Ijiri, Y. 1993. The Beauty of Double-entry Bookkeeping and its Impact on the Nature of Accounting Information. Economic notes: Economic Review of Banca Monte dei Paschi di Siena, 22 (2), 265-285.

Inghirami, I. E. 2019. Accounting Information Systems: the Scope of Blockchain Accounting. Conference paper. *ITAIS and MCIS 2019: 13th Mediterranean Conference on Information Systems and 16th Conference of the Italian Chapter of AIS (Pavia).*

IRS. 2015. Publication 583 (01/2015), Starting a Business and Keeping Records. *Internal Revenue Service*. Available at https://www.irs.gov/publications/p583#en_US_201408_publink1000253188

ISO/IEC. 2010. Information technology — Open-edi reference model. *ISO/IEC International Standard 14662*. Third edition 2010-02-15.

ISO/IEC. 2015. Information Technology — Business Operational View — Part 4: Business Transaction Scenarios — Accounting and Economic Ontology. *ISO/IEC International Standard 15944-4.* Second edition 2015-04-01.

ISO. 2019. ISO/IEC CD 15944-21 [ISO/IEC CD 15944-21]. *ISO*. Available at https://www.iso.org/standard/78924.html

Jeffries, D. 2020. Why Everyone Missed the Most Important Invention in the Last 500 Years. *Hackernoon*. Available at https://hackernoon.com/why-everyone-missed-the-most-important-invention-in-the-last-500-years-c90b0151c169

Kuhn, T. S. 1996. The Structure of Scientific Revolutions. Chicago, United States of America: The University of Chicago Press.

Liu, M., Wu, K., Xu, J. J. 2019. How Will Blockchain Technology Impact Auditing and Accounting: Permissionless versus Permissioned Blockchain. *Current Issues in Auditing*, 13 (2), A19-A29.

Lomax, P. S. 1918. Significant Results of Missouri and New Mexico Commercial Education Studies. *The School Review: A Journal of Secondary Education*, 26 (2), 73-84.

Mainelli, M., and Smith, M. 2015. Sharing Ledgers for Sharing Economies: an Exploration of Mutual Distributed Ledgers (aka Blockchain Technology). *The Journal of Financial Perspectives*, 3 (3), 38-58.

McCarthy, W. E. 1979. An Entity-Relationship View of Accounting Models. *The Accounting Review*, 54 (4), 667-686.

McCarthy, W. E. 1980. Construction and Use of Integrated Accounting Systems with Entity-Relationship Modeling. *Proceedings of the 1st International Conference on the Entity-Relationship Approach to Systems Analysis and Design*. In *Entity-Relationship Approach to Systems Analysis and Design*, edited by P. Chen, 625-37. Amsterdam: North Holland Publishing Company.

McCarthy, W. E., and Gal, G. 1981. Declarative and Procedural Features of a CODASYL Accounting System. *Entity-Relationship Approach to Information Modeling and Analysis*, 197-213. Washington: ER Institute, 1981.

McCarthy, W. E. 1982. The REA Accounting Model: A Generalised Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review*, 57 (3), 554-78.

McCarthy, W. E. 2001. Re: Company centric view ends. Data goes to the Net. Available at http://linas.org/mirrors/www.gldialtone.com/2001.07.14/McCarthyPost.htm

McCarthy, W. E. 2016. Accounting Is Big Data: How Research Becomes Reality. Accounting is Big Data Conference. *American Accounting Association*. Video. Available at https://aaahq.org/Meetings/2016/Accounting-Is-Big-Data-Conference/Video-Gallery/AIBD-Video-10-Public

McCarthy, W. E., and Holman, G. K. 2019. Blockchain and eBusiness. The Open-edi Distributed Business Transaction Repository. ISO/IEC 15944-21 OeDBTR." *E-Invoicing Exchange Summit Singapore 2019*.

Mohanty, D. 2018. Blockchain From Concept to Execution: BitCoin, Ethereum, Quorum, Ripple, R3 Corda, Hyperledger Fabric/SawTooth/Indy, MultiChain, IOTA, CoCo. New Delhi, India: BPB Publications.

Mohanty, D. 2019. R3 Corda for Architects and Developers. Noida, India: Apress.

Nakamoto, S. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Available at https://bitcoin.org/bitcoin.pdf

Nakamoto, S. 2009. Satoshi Reply to Mike Hearn. *Nakamoto Studies Institute*. Available at https://nakamotostudies.org/emails/satoshi-reply-to-mike-hearn/

Odom, C. 2013. Triple-Signed Receipts. *Open-Transactions*. Available at https://opentransactions.org/wiki/Triple-Signed_Receipts

Odom, C. 2015. Open-Transactions: Secure Contracts Between Untrusted Parties. Open-Transactions White Paper. Available at http://www.opentransactions.org/open-transactions.pdf

Pacio. 2018a. Pacio Technology. *Pacio Whitepaper Extract*. Available at https://www.pacio.io/docs/Pacio_Technology.pdf

Pacio. 2018b. Semantic Blockchain for Business Data: Competition. *Pacio*. Available at https://www.pacio.io/docs/PacioCompetition.pdf

Pacio. 2020. Pacio White Paper. *Pacio*. Available at https://pacio.io/docs/PacioWhitePaper.pdf

Pacioli, L. 1494. Particularis de Computis et Scripturis. In Pacioli, L. *Summa de Arithmetica Geometria Proportioni et Proportionalita*. English translation (1995). Seattle, United States of America: Pacioli Society.

Perez, D., Xu, J., and Livshits, B. 2020. Revisiting Transactional Statistics of High-scalability Blockchains. *IMC'20: Proceedings of the ACM Internet Measurement Conference (IMC).*

Peters-Richardson, J. 2011. Introduction to Accounting. *Ministry of Education, Trinidad and Tobago Commonwealth of Learning*. Available at http://oasis.col.org/bitstream/handle/11599/2463/2011_VUSSC_Intro-Accounting.pdf

Postma, J., and Helm, A. J. v. d. 2000. La Riegola de Libro: Bookkeeping Instructions From the Mid-fifteenth Century. *Accounting and History: a Selection of Papers Presented at the 8th World Congress of Accounting Historians*: Madrid-Spain, 19-21 July 2000, 147-178. Asociación Española de Contabilidad y Administración de Empresas, AECA.

Rao, V. S. 2020. Blockchain Accounting in A Triple Entry System – Its Implications on the Firm. *Our Heritage*, 68 (1), 10499-10512.

Request. 2018a. The concept of smart financial audits on the Blockchain. *Request Yellow Paper*. Available at https://request.network/assets/pdf/request_yellowpaper_smart_audits.pdf

Request. 2018b. The Future of Commerce: A Decentralised Network for Payment Requests. *Request Whitepaper*.

Rukhiran, M., and Netinant, P. 2018. The House Bookkeeping Conceptual Framework for Supporting Adaptability Using Three Dimensions Layering. *Advanced Science Letters*, 24 (7), 5234-5238.

Sangster, A. 2015. The Earliest Known Treatise on Double Entry Bookkeeping by Marino de Raphaeli. *Accounting Historians Journal*, 42 (2), 1-34.

Sangster, A., and Rossi, F. 2018. Benedetto Cotrugli on Double Entry Bookkeeping. *De Computis, Revista Española de Historia de la Contabilidad*, 15 (2), 22-38.

Sleeter, D. 2014. Bitcoin is based on this writing from Todd Boyle in 2003. http://ow.ly/t5axF We collaborated during his development. #roundtable14. *Twitter Post*. Available at https://twitter.com/DougSleeter/status/428612589795704832

Smith, G. 2019. The Many Facts Pointing to Ian Grigg Being Satoshi. *Bitcoin.com*. Available at https://www.bitscoins.net/the-many-facts-pointing-to-ian-grigg-being-satoshi/

Swanson, T. 2015. A Blockchain With Emphasis on the "A". *Great Wall of Numbers: Business Opportunities and Challenges in Emerging Markets*. Available at https://www.ofnumbers.com/2015/07/09/a-blockchain-with-emphasis-on-the-a/

Szabo, N. 1994. Smart Contracts. Available at http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Szabo, N. 1996. Smart Contracts: Building Blocks for Digital Markets. Available at http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html

Szabo, N. 2008. Bit Gold. *Unenumerated*. Available https://unenumerated.blogspot.com/2005/12/bit-gold.html

Tasca P., and Tessone C. 2019. Taxonomy of Blockchain Technologies. Principles of Identification and Classification. *Ledger,* 4.

Tsichritzis, D. and Klug, A. 1978. The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. *Information Systems*, 3 (3), 173-191.

Tyra, J. M. 2014. Triple Entry Bookkeeping With Bitcoin. *Bitcoin Magazine*. Available at https://bitcoinmagazine.com/articles/triple-entry-bookkeeping-bitcoin-1392069656

Valueflows. 2017. Transfer and Exchange Examples. *Valueflows*. Available at https://valueflo.ws/examples/ex-exchange.html

Venona Cypherpunks Archives. 2004. Cypherpunks Date Index for 1998 11. *Cypherpunks Archives*. Available at http://cypherpunks.venona.com/date/1998/11/

Victorovna, N. V. 2015. F. V. Ezersky and the Development of Accounting Thought and Practices in Russia. *Vestnik NSUEM*, 4, 160-170.

Vijai, C., Elayaraja, M., Suriyalakshimi, S. M., and Joyce, D. 2019. The Blockchain Technology and Modern Ledgers Through Blockchain Accounting. *Adalya Journal* 8 (2), 545-557.

Vollmer, H. 2003. Bookkeeping, Accounting, Calculative Practice: the Sociological Suspense of Calculation. *Critical Perspectives on Accounting*, 14 (3), 353–381.

Wang, Y.; Kogan, A. 2018. Designing Confidentiality-preserving Blockchain-based Transaction Processing Systems. *International Journal of Accounting Information Systems*, 30, 1-18.

Wild, J. J., Shaw, K. W., and Chiappetta, B. 2011. Fundamental Accounting Principles. New York, United States of America: McGraw-Hill/Irwin.

Wright, C. S. 2019. Dr. Craig Wright Explains the Origins of Bitcoin – Full interview (Edwards, J., interviewer). *Finder*. Available at https://www.finder.com/dr-craig-wright-explains-the-origins-of-bitcoin-full-interview

Zahnentferner, J. 2018. Chimeric Ledgers: Translating and Unifying UTXO-based and Account-based Cryptocurrencies. *IACR Cryptol. ePrint Arch*, 262-273.

APPENDIX A: PROFIT-CALCULATING SMART CONTRACTS

This appendix provides an illustrative example of smart contract usage to materialise different accounting approaches, as described in the main text.[32]

AVCO (Average Cost)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract AVCO {
struct Inventory{
    int value;
    int count;
}

/** creates an object type named "Inventory" with two properties:
total value and number of units */


Inventory inventory_1;
// initializes an inventory, which is initially empty (value=0, count=0)

int[] profits;
// creates a list (array) of all profits, which is initially empty

address payable inventoryHolder;
/** establishes that the contract writer (Alice= owns the inventory and that her address
(0x56..., defined as "inventoryHolder")
can receive Ether from any user interacting with this smart contract (in this case Bob) */

uint public itemPrice;
// declares the value of the item (in wei)

mapping (address => int) inventoryBalance;
/** creates a look-up table named "inventoryBalance" where an address is mapped one-to-one
to the corresponding balance */

constructor (uint _itemPrice) {
    inventoryHolder = msg.sender;
    itemPrice = _itemPrice;
    }
// establishes that when Alice deploys the contract, she sets the price of sale

function getInventory() public view returns(int, int) {
    return (inventory_1.value, inventory_1.count);
}
```

[32] This is potentially an Online Appendix.

```
// allows to obtain the value and number of items in the inventory

function getProfits() public view returns(int[] memory) {
return profits;
}
// allows to obtain the list of all profits (empty for the moment).

function getInventoryBalance(address someAddress) public view returns(int) {
return inventoryBalance[someAddress];
}
// allows to obtain the balance of an address inputted by the user as an integer

function addOneUnit(int inventoryAdded) public {
    require(inventoryHolder == msg.sender,"Only owner can add units to own inventory");
    inventoryBalance[inventoryHolder] += 1;
    inventory_1.value += inventoryAdded;
    inventory_1.count += 1;
}
/** allows the contract writer to add a new element to the inventory by specifying its value
The second line of this function is commented in the Kovan version to allow any user to test
the contract */

function buyOneUnit() public payable {
    require(inventoryHolder != msg.sender,"Owner cannot purchase own units");
    require(inventoryBalance[inventoryHolder] >0, "insufficient balance");
    require(itemPrice == msg.value, "Unmatched price");
    // allows the buyer to purchase a unit from the seller at the price established by her

    inventoryHolder.transfer(msg.value);
    /** allows the user interacting with the smart contract (Bob) to
    send a specific amount in wei (“msg.value”) to the inventory holder (Alice) */

    int cogs = inventory_1.value/inventory_1.count;
    // defines cost of goods sold (“cogs”) as the average cost of the items in the inventory

    inventoryBalance[inventoryHolder] -= 1;
    // redefines the seller’s (Alice’s) inventory as its previous value minus one

    inventoryBalance[msg.sender] += 1;
    // redefines the buyer’s (Bob's) inventory as its previous value plus one

    int profit = int(msg.value) - cogs;
    /** defines profit as the difference between sale price and cogs.
    Note that both value and cogs are measured in wei in this example */

    profits.push(profit);
    // includes the profit of this transaction in the list (array) of profits

    inventory_1.value -= cogs;
    inventory_1.count -= 1;
    // updates the inventory of the beginning of this contract to reflect this transaction
}
}
```

FIFO (First-in, First-Out)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract FIFO {

int[] inventory;
// initializes inventory balance with an empty array

int[] profits;
// creates a list (array) of all profits, which is initially empty

address payable inventoryHolder;
/** establishes that the contract writer (Alice= owns the inventory and that her address
(0x56..., defined as "inventoryHolder") can receive Ether from any user interacting with this
smart contract (in this case Bob) */

uint public itemPrice;
// declares the value of the item (in wei)

mapping (address => int) inventoryBalance;
/** creates a look-up table named "inventoryBalance" where an address is mapped one-to-one to
the corresponding balance */

constructor (uint _itemPrice) {
    inventoryHolder = msg.sender;
    itemPrice = _itemPrice;
    }

function getInventory() public view returns(int[] memory) {
    return inventory;
}
/** The user interacting with the smart contract can obtain the array of values of items in t
he inventory */

function getProfits() public view returns(int[] memory) {
return profits;
}
// returns the list of all profits (empty for the moment)

function getInventoryBalance(address someAddress) public view returns(int) {
    return inventoryBalance[someAddress];
}
// returns the balance an address inputted by the user as an integer

function addOneUnit(int inventoryAdded) public {
    require(inventoryHolder == msg.sender,"Only owner can add units to own inventory");
    inventory.push(inventoryAdded);
    inventoryBalance[inventoryHolder] += 1;
}
/** allows the contract writer to add a new element to the inventory by specifying its value
The second line of this function is commented in the Kovan version to allow any user to test
the contract */
```

```
function buyOneUnit() public payable {
    require(inventoryHolder != msg.sender,"Owner cannot purchase own units");
    require(inventoryBalance[inventoryHolder] >0, "insufficient balance");
    require(itemPrice == msg.value, "Unmatched price");
    inventoryHolder.transfer(msg.value);
    // allows the buyer to purchase a unit from the seller at the price established by her


    int cogs = inventory[0];
    // defines costs of goods sold (“cogs”) as the value of the first item in the inventory

    inventoryBalance[inventoryHolder] -= 1;
    // ” ’Alice’s inventory is redefined as its previous value minus one

    inventoryBalance[msg.sender] += 1;
    // ” ’Bob’s inventory is redefined as its previous value plus one

    int profit = int(msg.value) - cogs;
    // calculates profit of this sale as the difference between price and cogs

    profits.push(profit);
    // add an element into the `profits` array

for(uint i = 0; i < inventory.length - 1; i++) {
inventory[i] = inventory[i+1];
}
inventory.pop();
    // removes the first element in the “inventory” array
    }
}
```

LIFO (Last-in, First-Out)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract LIFO {

int[] inventory;
// initializes inventory balance with an empty array

int[] profits;
// creates a list (array) of all profits, which is initially empty

address payable inventoryHolder;
/** establishes that the contract writer (Alice= owns the inventory and that her address
(0x56..., defined as "inventoryHolder") can receive Ether from any user interacting with
this smart contract (in this case Bob) */

uint public itemPrice;
// declares the value of the item (in wei)

mapping (address => int) inventoryBalance;
/** creates a look-up table named "inventoryBalance" where an address is mapped one-to-one
to the corresponding balance */

constructor (uint _itemPrice) {
    inventoryHolder = msg.sender;
    itemPrice = _itemPrice;
    }

function getInventory() public view returns(int[] memory) {
    return inventory;
}
/** The user interacting with the smart contract can obtain the array of values of items in
the inventory */

function getProfits() public view returns(int[] memory) {
return profits;
}
    // returns the list of all profits (empty for the moment)

function getInventoryBalance(address someAddress) public view returns(int) {
    return inventoryBalance[someAddress];
}
// returns the balance an address inputted by the user as an integer

function addOneUnit(int inventoryAdded) public {
    require(inventoryHolder == msg.sender,"Only owner can add units to own inventory");
    inventory.push(inventoryAdded);
    inventoryBalance[inventoryHolder] += 1;
}
/** allows the contract writer to add a new element to the inventory by specifying its value
The second line of this function is commented in the Kovan version to allow any user to test
the contract */

function buyOneUnit() public payable {
    require(inventoryHolder != msg.sender,"Owner cannot purchase own units");
```

```
    require(inventoryBalance[inventoryHolder] >0, "insufficient balance");
    require(itemPrice == msg.value, "Unmatched price");
    inventoryHolder.transfer(msg.value);

    int cogs = inventory[inventory.length - 1];
    // defines costs of goods sold (“cogs”) as the value of the last item in the inventory

    inventoryBalance[inventoryHolder] -= 1;
    // ” ’Alice’s inventory is redefined as its previous value minus one

    inventoryBalance[msg.sender] += 1;
    // ” ’Bob’s inventory is redefined as its previous value plus one

    int profit = int(msg.value) - cogs;
    // calculates profit of this sale as the difference between price and cogs

    profits.push(profit);
    // add an element into the `profits` array

inventory.pop();
    // removes the last element in the “inventory” array
    }
}
```

APPENDIX B: ACRONYMS

ANSI/X3	American National Standards Committee on Computers and Information Processing
AVCO	Average Cost
B2B	Business-to-business
CLOCKSS	Controlled Lots of Copies Keep Stuff Safe
DLT	Distributed Ledger Technology
ERP	Enterprise Resource Planning
FIFO	First-in, First Out
G2G	Government to Government
GAAP	Generally Accepted Accounting Principles
GL	General Ledgers
IEC	International Electrotechnical Commission
ISO	International Organization for Standardisation
LIFO	Last-in, First-Out
LOCKSS	Lots of Copies Keep Stuff Safe
OeBTO	Open-edi business transaction ontology

OeDBTR	Open Electronic Data Interchange Distributed Business Transaction Repository
PTR	Public Transaction Repository
REA	Resource-Event-Agent
SPARC	Standards Planning and Requirements Committee
STR	Shared Transaction Repository
TEA	Triple-entry accounting
TEB	Triple-entry bookkeeping

APPENDIX C: ACKNOWLEDGEMENTS

We thank the following individuals for answering our questions per email correspondence, per social networks or per interview:

Individual	Position
Chris Cook	Senior Research Fellow at UCL
Chris Odom	Founder of Open-Transactions
Craig S. Wright	Chief Scientist at nChain
David Hartley	CEO of Pacio
G. Ken Holman	CTO at Crane Softwrights Ltd and Editor of ISO/IEC 15944-21
Ian Grigg	CTO at Solidus
Marcos Miranda	Blockchain Developer at Smat
Melisa Baudon	Lead Designer at Smat
Robert Haugen	Developer at Mikorizal Software
Todd Boyle	Founder of GL Dialtone
Vincent Piscaer	Head of Alternative Investments at EEA Fund Management Limited

Commenting, however, does not equate to endorsement of the views expressed in this paper.