frontiers in Blockchain

TECHNOLOGY AND CODE
published: 25 October 2019
doi: 10.3389/fbloc.2019.00013

Invoice Discounting: A Blockchain-Based Approach

Nadia Fabrizio [1], Elisa Rossi [2], Andrea Martini [2], Dimitar Anastasovski [1], Paolo Cappello [1], Lorenzo Candeago [3,4,5] and Bruno Lepri [3]*

[1] CEFRIEL, Milan, Italy, [2] GFT Italia Spa, Genova, Italy, [3] Fondazione Bruno Kessler, Trento, Italy, [4] SKIL (Semantic and Knowledge Innovation Lab) Joint Open Lab, TIM-Telecom Italia, Rome, Italy, [5] DISI-University of Trento, Trento, Italy

Invoice discounting is a market with a double-digit potential growth rate over the next years in Europe and worldwide. The main benefit of *invoice discounting* is the acceleration of cash flow from customers to suppliers: suppliers get advance payments from the bank rather than waiting for the customers to pay. Hence, thanks to the quick availability of capital, businesses can invest in expansion and growth. More specifically, one of the most relevant problems today is how to provide better and faster *invoice discounting* services while preventing double spending and maintaining risk low. The blockchain frameworks have the potential to provide the right solution and thus to revolutionize the *invoice discounting* process. The benefits for suppliers, customers and financial institutions are related to the increased transparency added to the whole discounting process and the following risk reduction for the banks due to the capability to enhance the entire process and to reduce the double spending. In our paper, we introduce a blockchain-based *invoice discounting* system, called Distributed Ledger Invoice, and we propose a novel assessment method for evaluating currently available blockchain solutions for the *invoice discounting* scenario. Moreover, we also discuss two main issues regarding the *information accessibility* and the *interoperability*. In particular, since blockchain is still an emerging technology *interoperability* is a key factor for blockchain's adoption in inter-banking processes, where different blockchain solutions might be used. In this work we propose a *decoupling layer*, based on the Attribute-Based Access Control language, to unify the access control to reserved information across heterogeneous blockchains.

Keywords: invoice discounting, inter-banking processes, blockchain assessment model, blockchain interoperability, Attribute-Based Access Control

OPEN ACCESS

Edited by:
Diego Valiante, University of Bologna, Italy

Reviewed by:
Hugo E. Benedetti, University of Los Andes, Chile
Qinghua Lu, Data61 (CSIRO), Australia
Beth Kewell, University of Exeter, United Kingdom

****Correspondence:***
Nadia Fabrizio
nadia.fabrizio@cefriel.com

Specialty section:
This article was submitted to Financial Blockchain, a section of the journal Frontiers in Blockchain

Received: *30 April 2019*
Accepted: *20 September 2019*
Published: *25 October 2019*

Citation:
Fabrizio N, Rossi E, Martini A, Anastasovski D, Cappello P, Candeago L and Lepri B (2019) Invoice Discounting: A Blockchain-Based Approach. Front. Blockchain 2:13. doi: 10.3389/fbloc.2019.00013

1. INTRODUCTION

Nowadays, *invoice discounting* represents 10% of banks' provided credit, and it has become a major source of working capital finance globally after the restriction of bank financing due to the 2011 credit crunch (Wehinger, 2013). In particular, *invoice discounting* helps companies, especially Small and Medium Enterprises (SMEs), that have cash flow problems because of late payments from customers (i.e., invoices are usually paid in 30–90 days). For European SMEs it has surpassed loans and other forms of financing over the past decade (Wehinger, 2013). Another advantage of *invoice discounting* is the *confidentiality*: namely, suppliers control the sales ledger by collecting payments as usual and sending out reminders. The customer (debtor) is not involved in the discounting process, hence it is not informed that the supplier (creditor) is getting his/her credit financed. From the point of view of the suppliers' companies, the *confidentiality* is an advantage since, for easing negotiations with partners, they might not want to disclose their use of working capital finance.

Invoice discounting is a typical financial business to business (B2B) process (EUF, 2014). The actual legacy banking system has been based on a central database paradigm, which implies having a trusted and secure single point of failure and also implies an informative asymmetry in the ecosystem. This asymmetry can be maintained as far as there is a part in the ecosystem that is accountable for the whole process. However, the market changes in finance today pose the challenge to rethink the way banks deliver their services and to develop new market models (Tank, 2018). Thus, the financial sector has started exploiting the decentralization since the financial crisis of 2011, and the potential of blockchain has already been studied in recent years (Treleaven et al., 2017).

For the discounter the benefits of decentralization and blockchain adoption are as follows: (i) an immutable and time stamped record of the existence of every invoice emitted by a company, (ii) an immutable and time stamped record of the debtor's receipt, and (iii) the confirmation and verification of the invoice (against which a discounter would fund). Hence, the overall *invoice discounting* process will be enhanced. Indeed, the trust and security mechanisms of the blockchain allow for the elimination of on-site audits of receivables and debtors, of receivables' notification and debtors' verification, and of month-end reconciliation processes. Moreover, the adoption of blockchain will also allow for a fast and cheaper value transfer, in particular for cross-border payments.

More specifically, the debtor's verification of the invoice validity and of the reception of goods and services reduces significantly the risk of dispute and non-payment of that invoice. Moreover, the debtor could have an incentive to acknowledge and confirm its invoices without delay, as his/her own track record of confirming invoices would be visible on the blockchain to his/her suppliers and thus it could be used to influence the payment terms and the offered contract prices. This immutable debtor's verification could also potentially eliminate the risk of invoice fraud for a discounter as there would be no "consensus" met for double invoicing transactions. In fact, time-stamping an invoice has a legal value: if a company attempts to assign its invoice more than once, it would prevent any subsequent assignee being a *bona fide* purchaser for value without notice, thus protecting the first assignee.

In our proposed *invoice discounting* system, two main aspects have been investigated: (i) the *information accessibility*, and (ii) the *interoperability*, in terms of capability of deploying an infrastructure relying on different blockchains.

Regarding *information accessibility*, the data are stored in the blockchain and can be searched, extracted and analyzed for as long as is desired by the parties. Upon request, authorized third parties could therefore be able to view the full transaction, the payment and the performance history of a company. Indeed, the blockchain provides a complete and transparent record of a supplier's completed transactions and their success rate (e.g., what percentage of goods/services are returned or rejected and for what reason) on which to ground funding and recourse decisions. Regarding the debtors, the blockchain provides a complete and transparent record of their payment history that can be used to evaluate decisions about credit limits and debtor limits.

As previously anticipated, the second important aspect taken into account by our system is *interoperability*. Since different bank consortia might use different blockchain solutions, a *decoupling layer* to unify access control to reserved information is needed. Moreover, different banks might have different access rules for various information: for example, depending on the subscription level, internal policies or privacy restrictions.

However, during the data sharing processes over the blockchain, the same access rules need to be enforced. Therefore, there is a clear need for a *decoupling layer* that allows to write access control rules and enforce them independently of the blockchain that the bank is using. In our *invoice discounting* system, we have proposed to use a language that manages the access control on different blockchains instead of a blockchain interoperability language (Hardjono et al., 2019), due to the complexity and maintenance costs of the latter type of language.

The remainder of the paper is organized as follows: in section 2 we describe the scenario, the business and the technical requirements identified for an *invoice discounting* system. In section 3 we discuss the evaluation criteria for the blockchain selection as well as the Attribute-Based Access Control language (Hu et al., 2014, 2015), chosen as a way to build a *decoupling layer* among different blockchain solutions. Then, the architecture of the Distributed Ledger Invoice (DLI) system is introduced in section 4, particularly focusing on the blockchain and decoupling layers. Section 5 describes the evaluation of the proposed DLI system. Finally, we discuss the proposed solution and we draw our conclusions in section 6.

2. SCENARIO DEFINITION AND REQUIREMENTS' ELICITATION

As a first step, we have ran a series of workshops with bank and blockchain experts in order to define the scenario targeted by our Distributed Ledger Invoice (DLI) system. Before describing the four phases composing the *invoice discounting* scenario, it is worth mentioning the parties involved in this process: (i) the *supplier*, the entity (i.e., person, company) asking for an *invoice discounting* to the bank, (ii) the *customer*, the entity (i.e., person, company) that has to pay the invoice, and (iii) the *bank*, the financial entity receiving and accepting/rejecting the *invoice discounting* request.

As anticipated above, the *invoice discounting* process is composed of four phases (see **Figure 1**): (i) a *generation* phase, where the supplier feeds the repository of his/her own e-invoice system with the invoice(s) of interest; (ii) a *publishing* phase, where the supplier visualizes through the system all the invoices

Abbreviations: DLI, Distributed Ledger Invoice; DLT, Distributed Ledger Technologies; BAM, Blockchain Assessment Model; ABAC, Attribute Based Access Control; XACML, eXtensible Access Control Markup Language; DAM, Digital Asset Management; TPS, transactions per second; PDP, Policy Decision Point; PEP, Policy Enforcement Point; PAP, Policy Administration Point; PRP, Policy Retrieval Point; PIP, Policy Information Point; MVP, Minimum Viable Product; UAT, User Acceptance Testing; B2B, Business-to-Business; B2C, Business-to-Consumer; API, Application Programming Interface; JSON, JavaScript Object Notation.

FIGURE 1 | Portion of the Business-to-Business invoice discounting process: the blue arrows indicate the steps covered by the DLI System.

of the e-invoice system's repository and he/she selects the invoices to enter in the DLI system (once the invoice is published, the system assigns automatically a unique ID to it); (iii) an *evaluation* phase, where the bank receives automatically the request from the supplier and, after its internal assessment and authorization procedures, decides to accept or reject the invoice discounting. Then, the bank updates the system both in case of acceptance and rejection (in case of acceptance the system records the discounted percentage); and finally (iv) an *updating (acceptance/refusal)* phase, where the system allows the supplier to monitor in real time the status of the request. The other banks can visualize if an invoice has been discounted or not, and in which percentage, but without any reference to the bank involved.

Starting from this scenario, a set of *business requirements* has been identified during the workshops and organized in four main categories: (i) the *solution*, (ii) the *data security*, (iii) the *interoperability*, and (iv) the *profiling*.

Regarding the *solution* category, the whole blockchain-based *invoice discounting* process has to be faster than the current one, thus leading to a cost reduction. Moreover, the *solution* has to be user-friendly, while it is not yet necessary an integration with the informative systems of the bank. Again, the *solution* has to provide a control system to avoid double spending. Finally, the proposed *solution* has to provide internal private storage that records all the inserted invoices, in order to track the whole "history" of each invoice and to allow banks to search for information about the invoices for which a discount request has been opened.

Regarding the *data security* category, the treatment of the invoices' data has to respect the regulations about sensitive data, and the data should be inserted and shared following the *data minimization* principle (Article 5 of the Directive 95/46/EC-General Data Protection Regulation[1]).

Then, the last two key *business requirements* of the solution are *interoperability* and *profiling*. Thus, the system has to provide a layer to allow the connection between services implemented with different blockchain technologies (*interoperability* requirement) as well as to share with banks useful data and details to rate suppliers and customers in case of new requests (*profiling* requirement).

As final step of the workshops, the *technical requirements* were elicited from the *business requirements*. In particular, during the elicitation phase each technical requirement has been related to the part of the system that it affects, namely (i) the whole system, (ii) the decoupling layer, and (iii) the blockchain infrastructure.

Thus, the *system* should allow the supplier (i) to store his/her own invoices in a persistent way, (ii) to select the invoices for the submission and for the visualization, (iii) to request discounts for the submitted invoices to a bank part of the DLI network, and (iv) to monitor in real-time the status of his/her requests. The *system* should also be able (v) to automatically notify the bank for incoming requests of the supplier as well as (vi) to provide to the bank a way for accepting/rejecting the supplier's requests. Then, the *system* should be able (vii) to automatically notify the supplier about the bank's acceptance/rejection of his/her submitted invoices, (vii) to record the accepted discount amount once an invoice is accepted by the bank and (viii) to allow other banks to check if an invoice has been discounted or not and in which percentage, without any reference to the involved bank.

The *decoupling layer*, instead, should permit the DLI system to switch from different blockchain back ends in order to respond to market demands and technological changes and advancements as well as to adapt to different access rules of the banks, such as subscription levels, internal policies and privacy restrictions.

Finally, the *blockchain infrastructure* should (i) provide permission and access as well as (ii) guarantee the confidentiality of the exchanged information in order to accomplish the rules regarding privacy and data protection. Moreover, the *blockchain infrastructure* (iii) should allow standard administrative accountability in order to satisfy the financial requirements, (iv) should provide management of tokens and assets to deal with the digital invoices and the process of discounting, (v) should support up to 100 geographically distributed nodes, (vi) should provide a transaction speed and a volume capacity adequate to the financial scenario, and (vii) should be mature for product deployment and then ready for the go-to-market phase.

It is worth noting that the requirements related to the blockchain infrastructure have been weighted in order to correctly balance the metrics for the assessment of blockchain solutions evaluated in section 3.

3. METHODOLOGY

3.1. Evaluation Criteria for Blockchain Selection

The adoption of blockchain frameworks has a great potential in banking, but poses challenges related to the technological aspects (Puthal et al., 2018), to the impact (Vranken, 2017), and to the current regulations (Cocco et al., 2017). Hence, the

[1]https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679rom=EN

design of novel blockchain-based solutions, such as our DLI system, is a non-trivial task and requires a careful selection of the blockchain framework.

For defining the framework of DLI, we have developed an evaluation model, called Blockchain Assessment Model (BAM), that has been used to select the adequate blockchain framework for our *invoice discounting* scenario. More precisely, BAM is a model based on seven major general criteria (see the following subsections for a detailed description). These criteria were identified starting from a generic taxonomy for blockchains, recently proposed by Tasca and Tessone (2019) and from the analysis of challenges available in literature (ul Hassan et al., 2019).

In particular, we have selected the criteria more relevant for our scenario and for the business and technical requirements described in the previous section. Moreover, these criteria, due to the need to propose a prototype able to be really deployed, were also combined with the "maturity model" for blockchains (Wang et al., 2016), consisting of five stages, namely *initial* (i.e., the chaotic starting stage of a new service/product), *repeatable* (i.e., the stage where some experiences of a service/product are derived by similar services/products), *defined* (i.e., the stage where a service/product is standard and well-documented), *managed* (i.e., the stage comprising the metrics for a qualitative evaluation of a service/product), and *optimizing* (i.e., the stage where a service/product is continuously improved). Indeed, the selected blockchain has also to demonstrate a sufficient maturity level in order to be considered trustworthy and reliable. Currently, we have a multitude of blockchain frameworks with solutions that provide many interesting features, but most of the time are not working as expected. Thus, these solutions are good for the development of prototypes but they are not adequate for final products where everything should work properly.

In particular, we have used BAM to evaluate a set of blockchain frameworks for the *invoice discounting* scenario, namely (i) HyperLedger Fabric[2], (ii) R3 Corda[3], (iii) Stellar[4], and (iv) MultiChain[5]. It is worth noting that although this model has been designed for a specific scenario, it can also be slightly revised to become a generic evaluation model for different application scenarios of blockchain technologies. Interestingly, similar models have been recently proposed for evaluating blockchain technologies for generic purposes (Trump et al., 2018) and for supporting specific public sector services, such as the German asylum process (Fridgen et al., 2018).

In the following subsections we describe in detail the seven major criteria composing our Blockchain Assessment Model (BAM).

3.1.1. Ecosystem Governance

The blockchain ecosystem can be *permissionless* (public), where everyone can see and interact with the ledger, and *permissioned* (private), where certain rules of privacy are endorsed. In particular, the *invoice discounting* scenario is a closed one: only certified entities (e.g., banks) can join and participate in the network. Therefore, it is fundamental that the selected blockchain is a *permissioned* one, or rather an *invitation-only* system.

Hence, we assigned a minimum score (1) if the blockchain under analysis does not permit the creation of a *permissioned* network and a maximum one (5) if the blockchain under analysis fully provides the creation of a *permissioned* network.

3.1.2. Administrative Accountability

In the *invoice discounting* scenario, the banks participating in the network should be authorized to access all the information regarding users' status and their previous applications for the discount of specific invoices. This is needed due to the fact that specific users' requirements have to be verified by the banks in order to grant the discounting request. Therefore, the chosen blockchain framework should allow a *supervisor role*, namely someone who has the permissions to read and check all the transactions on the blockchain, regardless of their possible confidentiality. Please note that read-only permissions on the transactions are required.

Thus, the evaluation model assigns the minimum score (1) if the blockchain does not provide the possibility of having a supervisor administrator, while it assigns the maximum one (5) if the blockchain has the possibility of having a supervisor administrator.

3.1.3. Confidentiality of Exchanged Information

Information and data exchanged over the network should be private and secured. This is fundamental due to the nature of the financial data shared across the involved parties. Normally, the confidentiality requirement is limited to the assurance that only authorized actors may access the system and that the confidentiality of a single transaction is not required. In our scenario, user data have to be protected and not disclosed over the entire life cycle of the *invoice discounting* process. Moreover, the system has to be capable to track the access and usage of data at any time.

It is worth noticing that *permissioned* (private) blockchain frameworks can provide different levels of a transaction's confidentiality, from the simple assurance that only authorized users may access the system to more sophisticated encryption systems that ensure only the sender and the recipient of a specific transaction may see its content, while the other parties know only about its existence.

Hence, our evaluation assigns a minimum score (1) if all the users are able to see the information contained within the blockchain under analysis and the complete set of confidential details carried out by the process, while it assigns a maximum score (5) if the transactions' details are only visible to the parties involved in the process and the data within the blockchain are secured and private.

3.1.4. Scalability and Performance

Since the *invoice discounting* is a business-to-business (B2B) scenario and not a business-to-consumer (B2C) scenario, the chosen blockchain technology should be able to scale up to a reasonable number of nodes (i.e., in the order of hundreds).

[2]https://www.hyperledger.org/projects/fabric
[3]https://www.r3.com/platform/
[4]https://www.stellar.org/
[5]https://www.multichain.com/

In our evaluation scale, we assign a minimum score (1) if the blockchain under analysis does not scale more than one hundred nodes and a maximum score (5) if the blockchain is able to scale to the required number of nodes for the *invoice discounting* scenario.

3.1.5. Transactions' Speed and Volume

In the *invoice discounting* scenario, the system has to process as many transactions as the technology allows in almost real time. However, transactions' speed can vary due to the number of participants (i.e., banks) in the network. Hence, it is very important, once there is an increased workload, to keep a stable speed.

Thus, the following scale has been used to evaluate this criterion: we assign a minimum score (1) if the technology under analysis takes more time than a traditional process, while we assign a maximum one (5) if the technology is processing each transaction in almost real time.

3.1.6. Crash Tolerance

Crash tolerance is a mandatory feature required by every distributed network, including blockchains. The information contained within the ledger has to be resilient to the failure of a sufficient number of nodes participating in the network. The maximum number of failures should be high enough to guarantee that the possibility for that catastrophic event to occur is virtually impossible.

Even for the *invoice discounting* scenario, the chosen technology must be resilient to crash and node failures. Therefore, this criterion has been assessed as follows: we have assigned the minimum value (1) if the blockchain under analysis is not crash-tolerant and does not guarantee the persistence of valuable information, while we assigned the maximum value (5) if the blockchain guarantees the persistence of all the valuable information for the *invoice discounting* scenario.

3.1.7. Assets' Management

Some blockchain frameworks provide a native way to manage customized assets over the network of nodes. An *asset* is a token that usually represents something which does not derive its value directly from the chain. A concrete example of assets' usage in the *invoice discounting* scenario would be a bank issuing an asset to a blockchain in order to represent the amount of cash the bank is holding.

In our evaluation model, we assign the minimum score (1) if the evaluated blockchain does not provide a native way for the management of assets, while we assign a maximum score (5) if the evaluated blockchain does provide a native way for managing the assets on the chain.

It is worth highlighting that our evaluation model has not taken into account (i) the environmental sustainability of the blockchain framework (i.e., energy consumption) and (ii) the ability of the blockchain framework to revoke or amend a transaction (i.e., in our invoice discounting scenario the ability of revoking a signature and amending/updating prices, payment terms, etc.). This is due to the fact that these two additional criteria are critical for *permissionless* (public) blockchains (Puthal et al., 2018) but not for the *permissioned* (private) blockchains we are evaluating.

Indeed, *permissioned* blockchains usually consist of a limited number of nodes and thus might implement consensus protocol layers that are sustainable in terms of energy consumption (Vranken, 2017). Again, regarding the ability to revoke transactions, MultiChain and other *permissioned* approaches provide the possibility of going back in the chronology of the network and modifying a transaction if all the nodes agree on this change (Davradakis and Santos, 2019). Instead, this is not possible with *permissionless* blockchains (Davradakis and Santos, 2019).

3.2. Blockchain Interoperability and Attribute-Based Access Control

As previously said, since different bank consortia might use different blockchain solutions, we need to design a framework that permits these solutions to operate together. Currently, there are some ongoing attempts to define a standard and an interoperability architecture, such as the International Standards Organization (Deshpande et al., 2017), the World Wide Web Consortium (W3C) with Interledger[6] focusing on the banking sector and aiming to create a blockchain-agnostic payment and money transferring solution, the IEEE Blockchain group (Blockchain Group, 2018) and a recent research proposal by the MIT Connection Science initiative (Hardjono et al., 2019). For a complete description of the different types of blockchain interoperability see Buterin (2016) and Hyland-Wood and Khatchadourian (2018).

However, there are very few prototypical solutions for blockchain interoperability, such as the one proposed by the R3 consortium (Buterin, 2016) or the one proposed by Accenture (2018). These solutions are still at the stage of designing the standards or the initial prototype, hence they are not usable in a production environment.

In our paper, we have proposed a different approach: a *decoupling layer* based on the Attribute-Based Access Control (ABAC) language (Hu et al., 2014, 2015). ABAC is a standard and well-established language for managing access control based on the attributes of the entities involved. Attributes can be related to the subjects that require access to a resource (e.g., a user, an application, etc.), to the action that the subjects want to perform (e.g., read a file), to the resources (e.g., a file, a database record, etc.) and to the environmental information (e.g., the time of the day, the machine from which the user is connected, etc.) (Hu et al., 2014, 2015). In the recent literature, ABAC and blockchains have been combined together for Digital Asset Management (DAM) (Zhu et al., 2018), e-Health (Dias et al., 2018), and Internet of Things (IoT) (Ouaddah et al., 2016; Dukkipati et al., 2018) applications.

ABAC rules are expressed in an eXtensible Access Control Markup Language (XACML) (Rissanen, 2013), a XML-based standard that supports Boolean logic for combining attributes and for writing the rules.

[6]https://interledger.org/

An example of an ABAC rule (in pseudo-code) is the following one:

IF User = Bank's Operator **AND** Request State = In Evaluation
AND Bank's Subscription Level ≥ Subscription Level for reading Rating/Report **THEN** Permit

We have chosen to define common access rules (based on ABAC) and have added a layer that enforces these access rules when writing, reading and storing data in different blockchains.

The layer allows for the various banks to write access control rules that are general and not specific to a particular blockchain. Hence, it becomes possible combining access control rules from different banks or bank consortia.

4. DISTRIBUTED LEDGER INVOICE (DLI) SYSTEM

4.1. High-Level Architecture

The high level architecture of our *invoice discounting* system, the so-called Distributed Ledger Invoice (DLI) system, is depicted in **Figure 2**. This architecture has been designed taking into account the business and technical requirements described in section 2.

More specifically, the components of the high-level architecture are (i) the *blockchain* layer, i.e., the infrastructure that enables data sharing, (ii) the *decoupling layer*, which is responsible for managing the access rules to the resources stored in the *blockchain*, (iii) the *back end* that contains the core business logic and finally the (iv) *front end* that provides the final users the interface to access the functionalities of the DLI system. Finally, an *internal database* is associated with the *back end* to store all the data that are not shared among the participants (e.g., banks' private data, such as customers' data).

In this paper, we focus on the two more novel layers of the solution we have developed: (i) the *blockchain layer* and (ii) the *interoperability layer*. We do not describe more standard components, such as the *back end* and the *front end*.

The *blockchain layer* and the *interoperability layer's* components are deployed in a unit called *node*: each bank owns a node, while the *administrative node* is owned by the provider of the DLI system. Indeed, the *administrative node* is the guarantor of the system, and it has the maximum trust level among all the participants. Moreover, it provides the addresses of the bank nodes and the credit scoring value as aggregate data to the participants (i.e., banks).

It is worth underlining that the developed DLI system has to be considered a Minimum Viable Product (MVP). However, the system's high-level architecture has been built in order to require minimum effort for the integration with current banks' workflows and processes. Indeed, from an architectural point of view, the DLI system can be integrated with the bank legacy system thanks to a *blockchain layer* built upon standard APIs (i.e., JSON APIs). Moreover, the *blockchain layer* is installed in the *back end* and thus the bank employees do not need specific training to adopt and work with the DLI system.

4.1.1. Blockchain Layer

Following the criteria described in section 3.1, MultiChain has been chosen over a set of four distinct blockchain frameworks, namely (i) HyperLedger Fabric, (ii) R3 Corda, (iii) Stellar, and (iv) MultiChain. In particular, MultiChain has several features that make it a very valuable choice for the development of the DLI system:

- *Rapid deployment*. MultiChain is a quick installing environment for deploying Minimum Viable Products

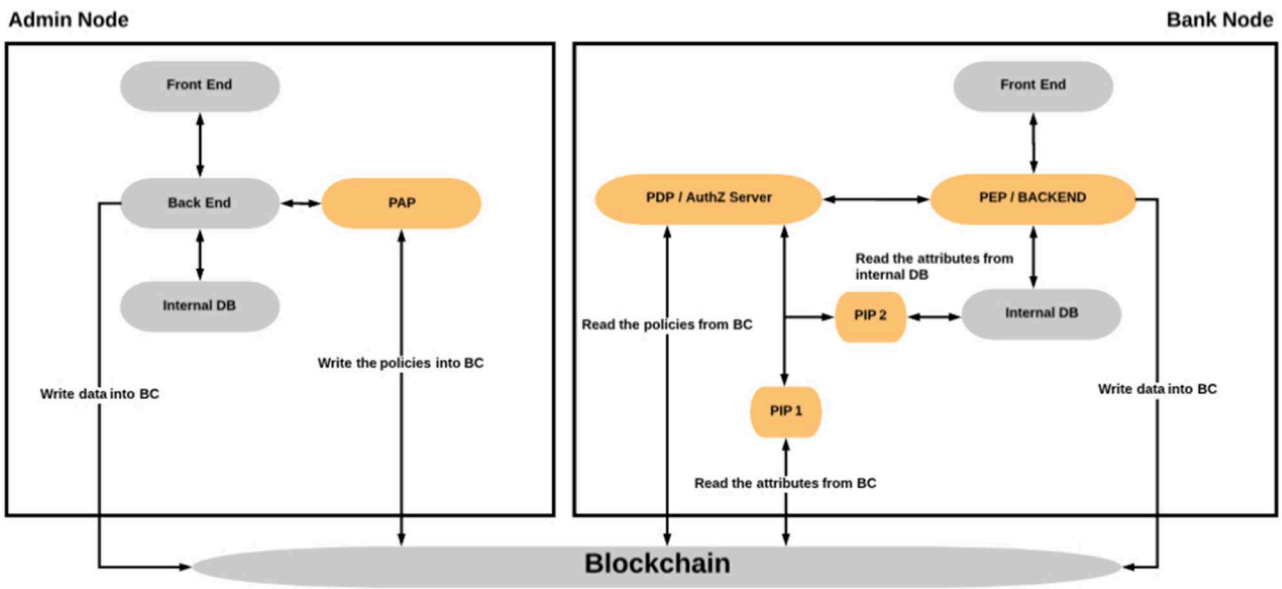

FIGURE 2 | A schema of the architecture of the Distributed Ledger Invoice system.

(MVP). Moreover, it can deploy unlimited blockchains per server for cross-chain applications.
- *Unlimited assets*. MultiChain permits the issue of millions of assets, all tracked and verified at the network level. In this way, it performs safe multi-asset and multi-party atomic exchange transactions.
- *Data streams*. MultiChain can create multiple key-values, time series or identity databases on a blockchain. Thus, it is ideal for data sharing, time-stamping and encrypted archiving.
- *Fine-grained permissions*. MultiChain can control who connects, send and receive transactions, create assets, streams and blocks. Hence, each blockchain is as open or as closed as the user needs.
- *Scalability*. MultiChain allows the block size to be adjusted and can handle up to 1,000 transactions per second, while *permissionless* (public) blockchains usually handle <10 transactions per second.

In the MultiChain blockchain layer of the DLI system, two types of assets are supported: (i) the *bank asset* and (ii) the *invoice asset*.

The *bank asset* represents the exchange unit of the bank. This asset is issued during the initialization step of a new participant bank (i.e., each bank that wants to join the network has to perform this step at first). The amount of the *bank asset* does not have an actual correspondence with the heritage of the bank, and this amount is usually a big enough number that the bank is permitted to accept up to 10 millions loans of 100% financing (this parameter is freely configurable by the bank). The *bank asset* ID is a unique string chosen by the bank.

The second one is the *invoice asset* that represents the published invoices in the DLI system: the issued raw units represent the full value of the invoice in percentage. The real amount of the invoice is encrypted as well as other invoices' sensible details. This information is stored in a special MultiChain stream. Moreover, the *invoice asset* ID is created by the DLI system using a hash function on the following parameters: (i) Invoice_Number, (ii) Supplier_Fiscal_ID, (iii) Invoice_Date, and (iv) Invoice_Value.

In addition to the *bank* and *invoice* assets, the DLI system uses the streams provided by MultiChain to store the data in a more structured way. Specifically, four streams are designed: (i) the *Assets*' stream, (ii) the *PubKeys* stream, (iii) the *Rating* stream, and (iv) the *Access* stream.

The *Assets*' stream contains the encrypted data stored during the exchange of assets. These data are used in two different phases, the *publishing* and the *request*. In the *publishing* phase the stored item represents the details of the invoice, which are encrypted using a symmetric key created by the publisher, while in the *request* phase the stored item represents the details of the request plus the symmetric key used to encrypt the details of the invoice (it is worth noting that this item is also encrypted with the public key of the recipient bank).

The *PubKeys* stream contains the couples "key user_address" and "value public_key." These items are stored in the stream during the *initialization* phase of the addresses. The participants adding a new item into the stream can update those public keys.

The *Rating* stream contains the couples "key user_fiscal_id" and "value single_rating." During the *acceptance/refusal* phase, the items are stored encrypted, with the public key of the *administrative node*.

Finally, the *Access* stream contains the access rules and the data used by the *decoupling layer* to decide. The items are stored in two different manners: (i) the access rules as a stream "key rule_name" and its value XACML and (ii) the setting data as a stream "key parameter_name" and its value.

In the *invoice discounting* scenario there are situations requiring the cancellation or modification of an already issued invoice. Thus, our DLI system has to manage these cases and record information regarding these changes in the blockchain layer. This is necessary in order to provide an updated status of the published invoices to all the participants. To this end, the MultiChain framework provides the possibility to "burn" an asset and make it no longer spendable. Furthermore, streams' ductility allows new data to be recorded and freely referenced to both new and preexisting assets.

Hence, our DLI solution proposes three different operational scenarios: (i) the cancellation of an invoice, (ii) the modification of one of the fields used for the creation of the *invoice asset* ID and (iii) the modification of something that does not concern one of the fields used for the creation of the *invoice asset* ID.

In case of *invoice cancellation*, the supplier has to start a cancellation process, which consists of the following two steps: (i) publishing a specific message containing the rectified *invoice asset* ID within the *Assets' stream*, and (ii) "burning" the quantity owned by the rectified asset. Furthermore, when an invoice has been canceled, the DLI system automatically notifies all the participants that own a quantity of the asset and are evaluating this specific *invoice discounting* request. The former have to "burn" the quantities they own; the latter have to reject the pending request.

In case of modification of one of the fields used for the creation of the *invoice asset* ID, the supplier has to start a *rectification* process, which consists of the following three steps: (i) publishing a new asset and its new data, (ii) publishing a specific message containing the rectified *invoice asset* ID and the new *invoice asset* ID within the *Assets' stream*, and finally (iii) "burning" the quantity owned by the rectified asset. In this third step, the DLI system and involved actors have to follow the same procedure as for the cancellation process. All the requests refused due to an invoice adjustment may be resubmitted with reference to the new asset.

Finally, in case of modifications not concerning the invoice fields used for the creation of the *invoice asset* ID, the *rectification* consists of the two following steps: (i) publishing the new data in the *Assets' stream* and (ii) publishing a specific message containing the rectified *invoice asset* ID within the *Assets' stream*. In this situation, it is not necessary to automatically refuse an *invoice discounting* request. Indeed, the evaluating bank is able to consider the new data to decide whether or not to grant the credit.

All modifications to the credit disbursement processes that have been already approved and the ones referring to canceled and/or rectified invoices must be managed outside the DLI system directly by the parties involved.

4.1.2. Decoupling Layer

As previously anticipated, we have proposed a *decoupling layer* based on the ABAC language (Hu et al., 2014, 2015) that allows for different banks to write and to combine general access control rules.

The architecture of the *decoupling layer* has the following ABAC modules:

- the Policy Enforcement Point (PEP) is responsible for the protection of the data on which ABAC rules are applied. More specifically, the PEP module has the role of evaluating a request and generating an authorization from this request. The authorization is then sent to the Policy Decision Point (PDP);
- the Policy Retrieval Point (PRP) retrieves and stores the deployed policies. Policies in ABAC are statements about attributes and they express what is allowed and what is not allowed;
- the Policy Decision Point (PDP) is the component which evaluates the incoming requests against the policies. This module returns a *permit/deny* decision. PDP may also use the Policy Information Point (PIP) to retrieve missing metadata;
- the Policy Information Point (PIP) bridges the PDP to external sources of attributes (e.g., in our DLI system to the blockchain);
- the Policy Administration Point (PAP) is the architectural entity used to manage policies. As we said before, these policies are later evaluated by the PDP.

In order to comply with the ABAC recommended structure, we have used the back-end of the bank as the Policy Enforcement Point (PEP) and we have introduced a new server for the capabilities of the Policy Decision Point (see the right side of **Figure 2**). The Policy Decision Point (PDP) uses a closely linked set of components, the Policy Information Points (PIPs), located on the same machine. PIPs talk directly with the data sources (e.g., the shared blockchains or the internal database) to obtain the attributes. Once a request (e.g., a new invoice in our scenario), sent by the Policy Enforcement Point (PEP), is successfully accepted from the Policy Decision Point (PDP) server, an authorized token is sent from the Policy Decision Point (PDP) to the Policy Enforcement Point (PEP), which is then allowed to write on the blockchain.

As regards the capabilities of the Policy Administration Point (the left side of **Figure 2**), the *administrative node* is responsible for the insertion of the policies in the blockchain. As further improvement, the bank can also have the capabilities of the Policy Administration Point (PAP). In this way we can have sub-policies for a more fine-grained control over the specific needs of a bank. Finally, access control policies are stored in the blockchain, making them tamper-proof and transparent.

5. SYSTEM EVALUATION

In order to evaluate the proposed DLI system, we have identified a specific scenario: the supplier (creditor) sends twice an *invoice discounting* request to the DLI system. In this way, the supplier (creditor) reaches 100% of the committed amount. Then, the same invoice is presented to a different bank as a paper invoice for an additional discounting request.

Starting from this scenario, the *evaluation phase* has taken place following the User Acceptance Testing (UAT) methodology (Cimperman, 2006). UAT has not only the goal of ensuring that a system does not crash and meets the technical requirements, but also that the system works for the stakeholder (Cimperman, 2006). Thus, we have directly involved the stakeholders, namely people from a group of Italian banks and from SIA (i.e., an Italian company that provides services and technologies for the banking sector). More precisely, the same people already involved in the business and technical requirements' elicitation phase as well as other people from the same banks and organizations were present in the testing sessions. These sessions have always been lead by the team developer manager.

The main achievements of the DLI solution, identified by the stakeholders, were as follows:

- *Invoice uniqueness*: the DLI system assigns a unique ID to each invoice, depending on the immutable parameters of the invoice. Hence, the invoices can be searched within the system through their IDs.
- *History of the invoice*: the DLI system records the information related to the uploaded and published invoices, including their history. For example, the system records if an invoice has already been discounted and the details related to the committed percentage.
- *Confidentiality of exchanged data*: the information and the data exchanged over the blockchain are private and secured, thus satisfying the strict confidentiality requirements of the financial entities involved in the transactions. Indeed, only the two parties involved in a transaction are able to see its content, while others may only verify if an invoice has already been published in the DLI system and the available amount for the discounting.
- *Decoupling layer*: our ABAC-based approach permits the interaction between the different blockchain-based solutions currently used by the stakeholders (i.e., banks).

6. DISCUSSION AND CONCLUSION

In our paper, we have provided evidence that the *invoice discounting* service might be improved by adopting approaches based on a distributed ledger. However, this improvement is possible only facing the challenges posed by the *information accessibility* and by the *interoperability*, in order to have a framework that is applicable in concrete environments and a technological approach able to sustain business-to-business (B2B) models in financial processes. Here, we have explored and selected a series of tools for defining a Minimum Viable Product (MVP) for the *invoice discounting* service according to the criteria of balancing the potential of the approach with the requirements of the financial sector. To this end, we have introduced a novel assessment model, called Blockchain Assessment Model (BAM), to evaluate the different blockchain

frameworks at disposal, and we have introduced the usage of the Attribute-Based Access Control language to overcome the interoperability issue.

More specifically, we have shown that the selection of the blockchain has to be based not only on the technological readiness (ISO, 2013) or on governance aspects (Trump et al., 2018), but also on other characteristics, such as the presence of *manageable assets*, *data privacy*, the *crash tolerance*, and the *scalability* properties. Indeed, we have articulated an assessment model based on seven criteria (see section 3.1), and we have applied our assessment model to the most widespread blockchain frameworks available for the financial sector (i.e., R3 Corda, HyperLedger Fabric, Stellar, and MultiChain). Our assessment model provides the benefit of having a single tool to define and select the relevant aspects of a blockchain framework for a specific inter-banking scenario, such as the one related to the *invoice discounting*.

Interestingly, the Blockchain Assessment Model might be adapted to other scenarios and use cases, for instance to business-to-business (B2B) processes related to supply chains (i.e., tracking of physical goods).

Finally, we have proposed another novelty, namely a *decoupling layer*, based on the Attribute-Based Access Control language, to unify the access control rules to reserved information across heterogeneous blockchains. In this way, it permits solutions, based on different blockchain technologies, to operate together. As a possible next step of our effort, we would be applying the same architecture and the same tools to other inter-banking processes regarding factoring and credit financing in general.

AUTHOR CONTRIBUTIONS

All the authors have designed the Distributed Ledger Invoice system. More specifically, NF, DA, and PC designed and applied the Blockchain Assessment Model (BAM). LC and BL proposed and implemented the Attribute-Based Access Control as interoperability framework. ER and AM worked on the definition of the business requirements as well as on the design of the architecture and the development of the final solution. All the authors have contributed to writing the paper and have approved the final submitted manuscript.

FUNDING

The work described in this article has been produced within the EIT Digital 2018 project Distributed Ledger Invoice (DLI). LC was supported by a fellowship from TIM-Telecom Italia, SKIL (Semantic and Knowledge Innovation Lab) Joint Open Lab.

ACKNOWLEDGMENTS

The authors want to thanks colleagues from the Fondazione Bruno Kessler (i.e., Silvio Ranise and Alessandro Tomasi) for their support on the Attribute-Based Access Control language and colleagues from the CEFRIEL-Politecnico di Milano Blockchain Lab for having supported the activities during the whole project duration. They also want to thank colleagues from GFT Italy Spa for having supported them with the business requirements and for helping to lead the whole project (i.e., Maurizio Ferraris and Cinzia Rubattino).

REFERENCES

Accenture (2018). *Connecting Ecosystems: Blockchain Integration*. Available online at: https://www.accenture.com/t20181022T205253Z__w__/us-en/_acnmedia/PDF-88/Accenture-20180514-Blockchain-Interoperability-POV.pdf

Blockchain Group, I. (2018). *IEEE Blockchain Standards*. Available online at: https://blockchain.ieee.org/standards

Buterin, V. (2016). *Chain Interoperability*. R3 Research Paper.

Cimperman, R. (2006). *UAT Defined: A Guide to Practical User Acceptance Testing*. Upper Saddle River, NJ: Pearson Education.

Cocco, L., Pinna, A., and Marchesi, M. (2017). Banking on blockchain: costs savings thanks to the blockchain technology. *Fut. Internet* 9:25. doi: 10.3390/fi9030025

Davradakis, E., and Santos, R. (2019). *Blockchain, Fintechs and Their Relevance for International Financial Institutions*. European Investment Bank: Working Papers 2019/01.

Deshpande, A., Stewart, K., Lepetit, L., and Gunashekar, S. (2017). *Distributed Ledger Technologies/Blockchain: Challenges, Opportunities and the Prospects for Standards*. Overview report The British Standards Institution (BSI).

Dias, J. P., Reis, L., Ferreira, H. S., and Martins, Â. (2018). Blockchain for access control in e-health scenarios. *arXiv* arXiv:1805.12267. doi: 10.1007/978-3-030-17065-3_24

Dukkipati, C., Zhang, Y., and Cheng, L. C. (2018). "Decentralized, blockchain based access control framework for the heterogeneous internet of things," in *Proceedings of the Third ACM Workshop on Attribute-Based Access Control* (New York, NY: ACM), 61–69.

EUF (2014). *Factoring and Commercial Finance-EUF Whitepaper*. Whitepaper.

Fridgen, G., Guggenmoos, F., Lockl, J., Rieger, A., and Schweizer, A. (2018). *Developing an Evaluation Framework for Blockchain in the Public Sector: The Example of the German Asylum Process*. European Society for Socially Embedded Technologies (EUSSET).

Hardjono, T., Lipton, A., and Pentland, A. (2019). Toward an interoperability architecture for blockchain autonomous systems. *IEEE Trans. Eng. Manage.* doi: 10.1109/TEM.2019.2920154. [Epub ahead of print].

Hu, V., Ferraiolo, D., Kuhn, D., Friedman, A., Lang, A., Cogdell, M., et al. (2014). *Attribute-Based Access Control Guide to Attribute Based Access Control (ABAC) Definition and Considerations*. National Institute of Standards and Technology, NIST SP 800-162.

Hu, V., Kuhn, D., and Ferraiolo, D. (2015). Attribute-based access control. *Comput. J.* 48, 864–866. doi: 10.1109/MC.2015.33

Hyland-Wood, D., and Khatchadourian, S. (2018). A future history of international blockchain standards. *JBBA* 1:3724. doi: 10.31585/jbba-1-1-(11)2018

ISO (2013). *Space Systems—Definition of the Technology Readiness Levels (TRLS) and Their Criteria of Assessment*. Standard 16290. Available online at: https://www.iso.org/standard/56064.html

Ouaddah, A., Elkalam, A., and Ouahman, A. (2016). Fairaccess: a new blockchain-based access control framework for the internet of things. *Security Commun. Netw.* 9, 5943–5964. doi: 10.1002/sec.1748

Puthal, D., Malik, N., Mohanty, S., Kougianos, E., and Das, G. (2018). Everything you wanted to know about the blockchain: its promise, components, processes, and problems. *IEEE Consum. Electron. Mag.* 7, 6–14. doi: 10.1109/MCE.2018.2816299

Rissanen, E. (2013). *Extensible Access Control Markup Language (XACML) Version 3.0*. OASIS Standard. Available online at: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

Tank, E. P. T. (2018). *Study in Focues–Competition Issues in the Area of Financial Technology*. Onlne Site (Various).

Tasca, P., and Tessone, C. J. (2019). A taxonomy of blockchain technologies: principles of identification and classification. *Ledger* 4, 1–39. doi: 10.5195/ledger.2019.140

Treleaven, P., Gendal Brown, R., and Yang, D. (2017). Blockchain technology in finance. *Computer* 50, 15–17. doi: 10.1109/MC.2017.3571047

Trump, B. D., Florin, M., Matthews, H. S., Sicker, D., and Linkov, I. (2018). Governing the use of blockchain and distributed ledger technologies: not one-size-fits-all. *IEEE Eng. Manage. Rev.* 46, 56–62. doi: 10.1109/EMR.2018.2868305

ul Hassan, F., Ali, A., Latif, S., Qadir, J., Kanhere, S., Singh, J., et al. (2019). Blockchain and the future of the internet: a comprehensive review. *arXiv:1904.00733*.

Vranken, H. (2017). Sustainability of bitcoin and blockchains. *Curr. Opin. Environ. Sustain.* 28, 1–9. doi: 10.1016/j.cosust.2017.04.011

Wang, H., Chen, K., and Xu, D. (2016). *A Maturity Model for Blockchain Adoption*. Berlin; Heidelberg: Financial Innovation Springer.

Wehinger, G. (2013). Smes and the credit crunch: current financing difficulties, policy measures and a review of literature. *OECD J.* 2, 115–148. doi: 10.1787/fmt-2013-5jz734p6b8jg

Zhu, Y., Qin, Y., Zhou, Z., Song, X., Liu, G., and Chu, W. C.-C. (2018). "Digital asset management with distributed permission over blockchain and attribute-based access control," in *2018 IEEE International Conference on Services Computing (SCC)* (IEEE), 193–200.

Conflict of Interest: ER and AM were employed by the company GFT.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.