A Blockchain-based Customizable Document Registration Service for Third Parties

Pamella Soares, Raphael Saraiva, Iago Fernandes, Antônio Neto, Jerffeson Souza
Graduate Program in Computer Science – State University of Ceara (UECE)
Blockchain One
Fortaleza, Ceará
{pamella.soares, raphael.saraiva, iago.fernandes, toinho.neto}@aluno.uece.br
jerffeson.souza@uece.br

2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) | 978-1-6654-9538-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICBC54727.2022.9805500

***Abstract*—Blockchain has been relevant in the document management process, serving as a storage solution with the potential to guarantee the relevant requirements needed for any document storage and validation solution. However, due to the distributed nature of blockchain, we may face implementation difficulties and high operational costs, for example. To facilitate this process, we propose a customizable blockchain-based document registration service that makes it possible to create different types of generalized documents for various application domains and store them in one or more blockchains integrated in an Application Programming Interface (API).**

***Index Terms*—Document Storage, API, Blockchain**

I. Introduction

Document management consists of processes that ensure the production, storage, and correct use of company or government documents. This management process has been carried out via intense digitalization, enabling an important digital transformation that also has been accelerated by the pandemic of the new coronavirus. In this context, blockchain has been a prominent strategy to the document management process, serving as a storage solution with the potential to guarantee the requirements of integrity, authenticity, access control, transparency, and availability necessary for any document management solution. In short, blockchain can be understood as a distributed database, protected by cryptography and governed by a consensus mechanism [1].

There are already solutions in the literature for registering documents supported by blockchain. Nevertheless, those proposals address specific domains with pre-defined business rules, such as platforms for registering diplomas [4], and API for registering water dam inspections [3], among others. Furthermore, there is the difficulty of implementing the blockchain infrastructure because it is essentially a distributed system and has had a high cost of operation and maintenance, affecting the development from different entities [2].

Motivated by this context, we propose a blockchain-based Customizable Document Registration Service to provide and facilitate integration with pre-configured blockchains to generically register and validate documents by third-party applications from different domains. This service allows the setting of parameters for information storage, from creating customizable document templates to choosing one or more target blockchains for storage. The main contributions from this study are (i) the development of a solution to enable the registration and validation of documents supported by pre-configured multi-blockchains, and (ii) a generic *developer-friendly* API to different contexts and third parties.

II. Customizable Document Registration Service: An Architecture Overview

This proposal presents a document registration and validation service supported by blockchain technology, allowing the creation of customizable models of documents that can be stored in different blockchains. The implemented API has functionalities to manage blockchain documents by third-party platforms with diverse domains. The architecture is composed of three main layers, as presented in Fig. 1: (i) Application Layer; (ii) Service Layer; and (iii) Persistence Layer.

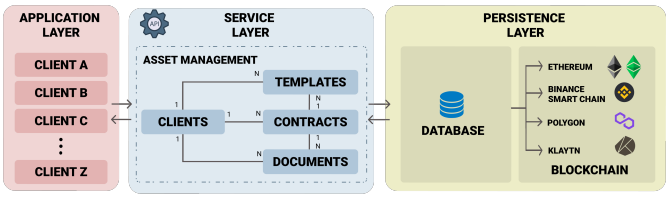

Fig. 1. Document Registration Service Architecture.

Regarding the *Application Layer*, we designed the API so that document creation is pervasive and can be integrated into any application domain that requires blockchain information storage and validation services. The Application Layer comprises all Clients that will consume the API, such as student diplomas in an undergrad course, the registration of news by the media press to combat Fake News, and several other application contexts.

The *Service Layer* deals with the implementation of API business rules regarding the functionalities that manage four main assets: (i) Client; (ii) Template; (iii) Smart Contract (SC), and (iv) Document. In short, the asset **Client** represents the entity coming from the Application Layer that will consume the service. Each Client can have 1 to N specific Templates, 1 to N SC, and 1 to N Documents. In turn, the **Template** is a structure of the Document type that the Client will add into the blockchain. A Template contains its respective structure attributes of a specific document and other information, like their descriptions. The asset **Smart Contract** is the deployed

contract that performs the main recording and reading functions for managing document information on the blockchain. Every SC can be associated with at least one previously registered Template, i.e., each SC stores Documents of one or more specific(s) Template(s).

To generalize the Templates across different records' domains, we designed the primary data structure in the SC that maps each vector index to a specific Template attribute. Moreover, the solution uses off-chain mechanisms from an external database to map each Template attribute with its respective blockchain content. In addition, we created a functionality in which the Client can use only one SC associated with several Templates. Finally, the asset **Document** consists of information content/value of a document to be managed based on the previously created Template.

The *Persistence Layer* uses a database as an off-chain mechanism and the blockchain for document information storage. Specifically, the use of a database has the primary objective of registering and managing Clients, storing their credentials, API access tokens, and other relevant information. Therefore, the database stores sensitive information and the attributes of templates created to perform data crossings between the database and the blockchain. The proposed service allows the insertion of information in one or more blockchains, allowing the Client system to choose the blockchain that best suits its demands. This requirement permits the Client to decide the storage based on how much it is willing to pay for fees; the speed of information recording, influenced by the consensus algorithm and technologies used by each blockchain; the reputation in the community; level of decentralization; among other considerations. The API-integrated blockchains follow an Ethereum-based framework, in which SC code is written in a stack-based bytecode language and executed on the Ethereum Virtual Machine; they are Ethereum, Ethereum Classic, Binance Smart Chain (BSC), Polygon, and Klaytn. Therefore, each public blockchain integrated into this solution can run the SC implemented through the Solidity language.

III. Workflow Demonstration

Fig. 2 presents, as an example, the registration flow of a student diploma information by a Client, an Undergraduate Course System. For this, we assumed that the Client has already registered with their proper access credentials to make requests to the API.

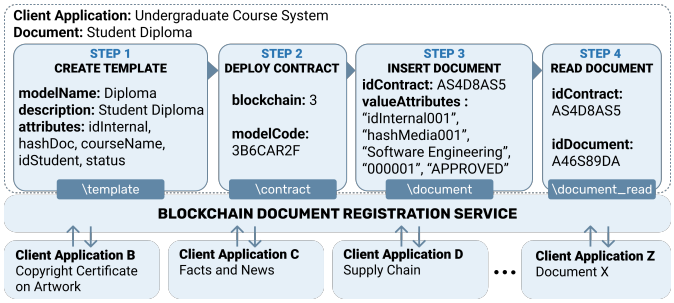

Fig. 2. Illustrative Flow of the API.

In **Step 1**, the Client creates a Template for the registration of students' Diplomas. We assumed the structure of the Diploma Template with the attributes: `courseName`, `idStudent` and a `status`. As an API rule, the Client must enter a `idInternal` reference to the identifier in the application database, and the `hashDoc`, consisting of the hash of the Document media to be inserted. When requesting `\template` with input attributes, the Client creates a new Template and receives a `modelCode`, the identifier of the Template created in the API and used for SC deployment. In **Step 2**, the Client requests the `\contract` to implement the SC on a pre-defined blockchain, indicating the one it wants to store the Diploma information. This way, the Client passes the blockchain code (`0:Ethereum`, `1:Ethereum Classic`, `2:BSC`, `3:Polygon` and `4:Klaytn`), and the `modelCode` that will be associated with the deployed SC. In this example, we will deploy the SC on the Polygon network, and this SC will only contain records referring to the Diploma Template. In turn, the API generates an `idContract` identifier when deploying an SC.

In **Step 3**, the API receives the `idContract` to identify the contract in which the Document information will be stored when requesting `\document`. At this point, the content of the document itself will be inserted, in this case, the diploma information indicates the student with `idStudent` '0001' from the Software Engineering (`nameCourse`) has a `status` of 'APPROVED'. Finally, this record can be consulted through the request `\document_read` at any time (**Step 4**). It is noteworthy that the Undergraduate Course System uses the API only to record the final information that the Client must properly curate. Furthermore, the Client can create templates with different attributes in its structure and request other functions to manage student diplomas, such as to write to more than one blockchain; check the media integrity through `hashDoc`, among others. We provided a visualization by Postman[1] of some requests shown in Fig. 2.

As future work, we intend to integrate a permissioned blockchain to define a more elaborate access control in case different Clients want to share sensitive information between them.

References

[1] Roman Beck, Michel Avital, Matti Rossi, and Jason Bennett Thatcher. Blockchain technology in business and information systems research. *Business Information Systems Engineering*, 59(10):381–384, nov. 2017.

[2] Xiaoyun Li, Zibin Zheng, and Hong-Ning Dai. When services computing meets blockchain: Challenges and opportunities. *Journal of Parallel and Distributed Computing*, 150:1–14, 2021.

[3] Antonio Jefferson Macedo, Allysson Allex Araújo, and Itamara Taveira. Adoção de blockchain para apoio ao cadastro e inspeção de barragens hídricas: Uma proposta de pesquisa baseada em design science research. In *5º Conferência sobre Sistemas de Informação na América Latina (ISLA)*, 2021.

[4] Raphael Saraiva, Allysson Allex Araújo, Pamella Soares, and Jerffeson Souza. Miriam: A blockchain-based web application for managing professional registrations of medical doctors in brazil. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–2. IEEE, 2021.

[1]https://zenodo.org/record/6140266