Enhancing the Reliability of Academic Document Certification Systems with Blockchain and Large Language Models

JEAN GILBERT MBULA MBOMA[1], OBED TSHIMANGA TSHIPATA[1], WITESYAVWIRWA VIANNEY KAMBALE[2,3], MOHAMED SALEM[2], MUDIAMPIMPA TSHYSTER JOEL[1], KYANDOGHERE KYAMAKYA[1,2]

[1]Génie Electrique et Informatique
Université de Kinshasa (UNIKIN)
H8J5+6PX Kinshasa
DEMOCRATIC REPUBLIC OF THE CONGO

[2]Institute for Smart Systems Technologies
Universitaet Klagenfurt
9020 Klagenfurt
AUSTRIA

[3]Faculty of Information and Communication Technology
Tshwane University of Technology
Private Bag x680, Pretoria, 0001
SOUTH AFRICA

Abstract: Verifying the authenticity of documents, whether digital or physical, is a complex and crucial challenge faced by a variety of entities, including governments, regulators, financial institutions, educational establishments, and healthcare services. Rapid advances in technology have facilitated the creation of falsified or fraudulent documents, calling into question the credibility and authenticity of academic records. Most existing blockchain-based verification methods and systems focus primarily on verifying the integrity of a document, paying less attention to examining the authenticity of the document's actual content before it is validated and registered in the system, thus opening loopholes for clever forgeries or falsifications. This paper details the design and implementation of a proof-of-concept system that combines GPT-3.5's natural language processing prowess with the Ethereum blockchain and the InterPlanetary File System (IPFS) for storing and verifying documents. It explains how a Large Language Model like GPT-3.5 extracts essential information from academic documents and encrypts it before storing it in the blockchain ensuring document integrity and authenticity. The system is tested for its efficiency in handling both digital and physical documents, demonstrating increased security and reliability in academic document verification.

Key-Words: Blockchain, Large Language Models, IPFS, Document Verification, Document Authentication, Reliability, SHA-256, Digital Signature

Received: August 27, 2023. Revised: July 9, 2024. Accepted: August 13, 2024. Published: September 25, 2024.

1 Introduction

Large Language Models and Blockchain may be two technologies with opposed approaches and enormous potential, but one thing seems certain: they both have the potential to transform our daily lives considerably and solve many problems whose solutions have hitherto remained incomplete or even unsatisfactory. The qualities of blockchain—decentralization, security, immutability, and transparency—have drawn the attention of numerous sectors since its inception in 2008, shortly after the paper on Bitcoin was published [1]. Notably, large language models (LLMs) like GPT-3 and GPT-4 have become quite strong tools for natural language processing in the last several years. These models have demonstrated an impressive ability to generate and understand content contextually and coherently while being remarkably versatile in a variety of natural language-related tasks [2]. Therefore, the combination of blockchain technology with LLMs presents the possibility of developing novel applications that capitalize on the advantages of both fields of technology. On the other hand, the steady increase in the number of falsified academic documents, whether digital or physical, has highlighted the limits of traditional verification methods [3], paving the way for the exploration of digital solutions better suited to contemporary challenges. Consequently, alternatives such as the digital signature [4], the QR code [5], [6], the barcode [7], [8], [9], and various other means [10], including the use of blockchain, have been implemented to this end. Speaking of the use of blockchain, it is undeniable that its characteristics of decentralization, immutability, and transparency have greatly complicated the task of counterfeiters; nevertheless, loopholes remain, and can lead to dramatic situations, especially in the case of document verification where the counterfeit could go unnoticed and be mistaken for the original. Most of the work and systems [3], [11], [12], [13], [14], [15], [16], [17], [18] [19], [20], [21] proposed to date have been based on the following assumptions:

1. The system (private or public blockchain) is considered very reliable (system reliability): fraud within the system (university, college, etc.) is underestimated.

2. Any document stored in the blockchain cannot be altered without obvious evidence of tampering (Document Integrity).

3. A document, subject to verification, is considered valid if it is present in the system, i.e., if its hash is stored in the blockchain. (Verification assumption).

These hypotheses are certainly relevant, but present certain vulnerabilities:

1. The system may be compromised to some extent; some validators may be dishonest or malicious.

2. Instead of directly modifying the original stored in the blockchain, the fraudster can manage to have a counterfeit validated, although this is not an easy task.

3. If a counterfeit manages to be validated in the blockchain without being detected, it will easily pass verification, which would be catastrophic.

It is therefore imperative to add a security mechanism to considerably tighten the validation process before registering a document. This would significantly increase the efficiency and reliability of document authenticity verification at a later stage.

2 Aims and Objectives

The main objective of our paper is to present a system in which Large Language Models are combined with blockchain technology to address this challenge. Our approach aims to use an LLM, GPT-3.5 in this case, to enhance the efficiency of document validation at registration and enable more reliable verification at a later stage; the idea being that if validation is much more rigorous, it is highly likely that documents registered on the blockchain are authentic and valid, and verification becomes simpler. When registering a document in our system, the LLM's role will be to extract the essential information from a document, whether it's a quotation sheet, a diploma, a certificate, a parcel document, or any other commercial or administrative document. This information will then be encrypted using a cryptographic hash function, specifically SHA-256. The result of this hash will then be recorded on the Ethereum blockchain, along with the document's content identifier (CID). This CID will be provided by the IPFS storage network we'll be using to host the document. It's important to note that the blockchain will mainly be used to store the CID and the hash generated by the LLM. Consequently, if we want to check the validity of any document possibly issued by our system, the latter will first extract the hash value associated with the key information of said document and compare this value with those of different documents present in the system. If there is a perfect match, the file under examination is certified authentic; if not, further analysis will be carried out to determine whether it is either dubious or simply missing from the system and apply the appropriate measures. In the following lines, we first outline the fundamental concepts related to LLMs, blockchain, and the IPFS decentralized storage protocol, while reviewing the work done in the context of document verification and authentication via blockchain. This is followed by a detailed presentation of our solution and the web platform used to implement our approach, as well as a discussion of the results obtained. At last, a conclusion is given in which prospects of improvement are also mentioned.

3 Background

3.1 Overview of LLMs

3.1.1 Introduction

Recently, there has been the emergence of sophisticated machine learning models capable of comprehending and producing writings in natural language. The Large Language Models (LLMs) refer to these. These models are usually trained on large and extensive textual datasets [22]. The development of these LLMs has been inspired mainly by the introduction of a neural network architecture called Transformers which enables them to mimic the complex structure of language and assists them in capturing long-range dependencies. This architecture enables them to learn and generate content with enhanced semantic richness and coherence. Like GPT-3 [23], PaLM [24], Galactica [25], and LLaMA [26], LLMs stand out due to their remarkable large-scale number of parameters, which can typically reach tens or hundreds of billions. A way to interact with these Large Language Models (LLMs) often involves prompt engineering, a method discussed in [23] and [27]. Prompt engineering involves users crafting precise and accurate prompts aimed at directing the LLMs to generate required outputs or performing certain tasks outlined by the input prompts. This systematic approach is widely used in contemporary evaluation methods, enabling humans to engage with LLMs through questioning or dialogue, essentially having conversations in natural language with these models. Table 1 provides a succinct yet essential comparison of conventional machine learning, deep learning, and LLMs.

In order to obtain a thorough understanding, let us examine the various language modeling approaches that have been used in the creation of LLMs thus far. Language modeling (LM) research has received a great deal of attention in the literature, as demonstrated by [28]. There are four major phases of development that this research may be divided into:

- **Statistical language models (SLM):** these emerged in the 1990s and are based on statistical learning. Building a word prediction model based on the Markov assumption—that is, predicting the next word based on the current context—is the fundamental idea. SLMs have been widely used to enhance task performance for natural language processing (NLP) [29], [30], [31], and information retrieval (IR) [32], [33]. However, because of the exponential number of transition probabilities that must be computed, high-order language models frequently suffer from the ”curse of dimensionality,” making accurate estimation of them challenging. Particular smoothing techniques, including back-off estimation and Good-Turing estimation, have been created to solve this issue and lessen the impact of sparse data. These techniques are intended to enhance the estimate of high-order language models and mitigate the problem of data sparsity.

- **Neural language models (NLM):** The foundation of NLM is neural networks, such as Recurrent Neural Networks (RNNs) [1], [34]. To predict the next word, these large language models (NLMs) rely on distributed word vectors, otherwise known as aggregated context features which capture the meaning of words based on their context in a sentence. This approach enables the efficient extraction of word vectors from an extensive textual dataset, using the capabilities of models like Word2Vec [35], GloVe [36], and FastText [37]. NLMs are also specifically designed to capture not only dense vector representations of individual words and sequences but also their long-term contextual dependencies. This makes well-trained NLMs synthesize coherent text through accurate next-word prediction based on prior context. Notably, the capabilities of these models extend beyond text generation, but extending various NLP tasks such as speech recognition, automatic translation, and a so many others [38].

- **Pre-trained language models (PLM):** These models become proficient at performing specific natural language processing (NLP) tasks after they’ve been trained on vast amounts of text input. This training also called pre-training involves exposing the model to large and varied sets of text, enabling it to mimic the structures, statistical patterns, and semantic relationships found within language. As a result of this comprehensive training, these language models can effectively predict and select the most likely next word in a sentence, drawing on their understanding of the context developed during training. Thanks to this approach, the model can capture different linguistic features and gain a sophisticated understanding of language. Typically, the pre-training phase is unsupervised, i.e. the model acquires knowledge from the data without the need for explicit annotations or labels.

After pre-training, PLMs can be fine-tuned for specific downstream tasks such as text classification, question answering, language translation, and other related applications. Fine-tuning is a process during which the model is trained on smaller, task-specific datasets containing labeled instances. In this way, the model adjusts its pre-training knowledge and abilities to function well on particular tasks. On the other hand, Pre-trained Language Models (PLMs) capture contextual dependencies in both directions, in contrast to Neural Language Models (NLMs) which predict the next word based on the previous context. PLMs consider both the words that come before and after the word they are predicting to comprehend its context properly. ELMo [39] is one example of this, as it uses a bidirectional LSTM network (biLSTM) for pre-training.

- **Large language models (LLM):** These are scaled-up versions of PLMs with, for the most

Table 1. Comparative study between traditional Machine Learning, Deep Learning, and LLMs (Source: [27])

Comparison	**Traditional ML**	**Deep Learning**	**LLMs**
Training Data Size	Large	Large	Very Large
Feature Engineering	Manual	Automatic	Automatic
Model Complexity	Limited	Complex	Very Complex
Interpretability	Good	Poor	Poorer
Performance	Moderate	High	Highest
Hardware Requirements	Low	High	Very High

part, billions or hundred billions of parameters. They are created by scaling PLMs (e.g., model size or data size). In fact, research has demonstrated that scaling PLMs frequently results in the model's increased capabilities and performance on downstream tasks [40]. Larger PLMs, such as the 175B-parameter GPT-3 and the 540B-parameter PaLM, have been shown to exhibit different behaviors during training than smaller PLMs, like the 330M-parameter BERT and 1.5B-parameter GPT-2, according to a study by [28]. These big PLMs are very good at difficult tasks because they have emergent skills that allow them to tackle complex task sequences. As a result, in the area of language modeling, AI algorithms have grown incredibly strong and efficient.

It is crucial to remember that an LLM is not always more capable than a small PLM, and certain LLMs might not exhibit emerging abilities. We can list the GPT-series (GPT-3, GPT-3.5, GPT4), Bard, Falcon, Llama, Bloom, and so on as examples of LLMs. An intriguing chronology of the major language models that have been in use recently is presented in the literature [28] (Fig. 1).

3.1.2 Emergent Abilities of LLMs

In the context of LLMs, emergent abilities refer to the unexpected or unplanned capabilities that the models display during training or application. These skills come from the models' exposure to enormous volumes of language data rather than being specifically programmed or taught to them. They are among the key characteristics that set LLMs apart from earlier PLMs [2]. It can be difficult to determine the essential size for emergent skills of LLMs because it varies based on the model or job being employed. This is the least size needed to possess a specific capacity. Three emergent skills that are typical for LLMs are introduced in the literature [28].

- In-context learning (ICL): Formally introduced by GPT-3, [23], it occurs when a model can complete a task using just a prompt made up of input-output samples. based solely on a prompt composed of input-output examples. Without any specific pre-training, the LLM can pick up knowledge from these examples [41]. The GPT-1 and GPT-2 models cannot be regarded as having the same level of ICL capacity as the GPT-3 model [28].
- Instruction following: On previously untested tasks that are also defined using instructions, LLMs have shown good performance through instruction tuning, which entails fine-tuning with a combination of multi-task datasets organized with natural language descriptions [42].
- Step-by-step reasoning: Unlike small language models, LLMs can tackle difficult tasks involving numerous reasoning steps (like mathematical word problems) using the chain-of-thought (CoT) prompting method. By using a prompting mechanism that includes intermediate levels of reasoning to arrive at the final answer, LLMs may handle problems of this nature [43].

3.1.3 Key Techniques for LLMs

Several key strategies are available to greatly increase the capacity of LLMs; we will quickly discuss and introduce a few here that have the potential to greatly increase LLM success [28].

- Scaling: As previously observed, the scaling laws show that expanding the model and dataset in addition to increasing the training computation often enhances the LLM's capabilities and performance [40], [44]. Furthermore, data scaling requires the use of a suitable cleaning procedure.
- Training: To successfully train a good LLM (known for its huge model size), distributed training algorithms are needed to learn efficiently the network parameters of LLMs. Many optimization frameworks, such as DeepSpeed, [45], and Megatron-LM, [46], have been made available to assist with the creation and deployment of parallel algorithms.

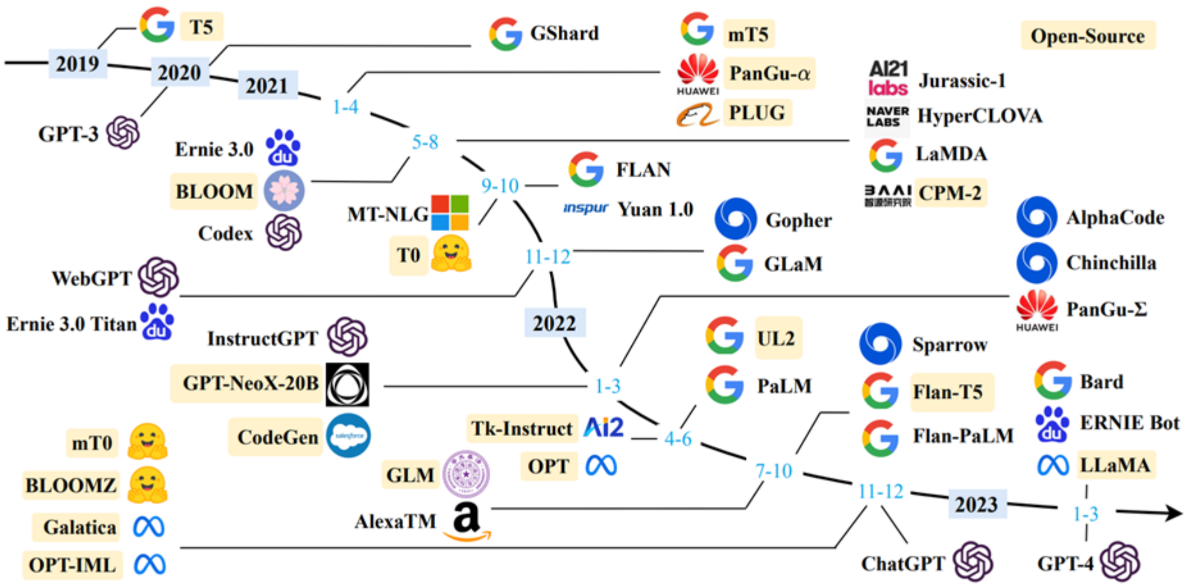


Fig. 1: A timeline of recent large language models (having a size larger than 10B) (Source: [28])

- Ability eliciting: After receiving extensive pretraining on large-scale corpora, LLMs may be able to solve common tasks. These qualities might not be apparent when carrying out particular jobs. To bring forth these skills, it is helpful to create task instructions that are relevant or in-context learning techniques. Strategies like instruction tailoring and Chain-of-Thought prompting can be used to improve the ability to generalize on untested tasks.

- Alignment tuning: Because LLMs are trained to identify the features of diverse data sets, they may produce biased or harmful material. The InstructGPT technique, [47], uses human input in conjunction with reinforcement learning, [47], [48], to steer LLMs in a manner consistent with human ideals. Using a similar methodology, ChatGPT exhibits a strong alignment capability by generating well-mannered and non-offensive responses, including declining to respond to inflammatory inquiries.

- Tools manipulation: LLMs are primarily trained to produce text from big corpora; nevertheless, they exhibit worse performance on tasks that are not text-based (numerical calculation). They are not able to capture current information and are only able to process training data. Using external tools such as a calculator, which can perform accurate calculations [2], and search engines, which can assist in locating unknown material [49], is one potential approach. Additionally, ChatGPT's usage of third-party plugins greatly increases LLMs' capabilities.

The key ideas of blockchain technology will be quickly covered in the following section after we have briefly examined the key facets of language models, in particular LLMs.

3.2 Overview of Blockchain

3.2.1 Blockchain network and structure

A distributed ledger that is decentralized is called a blockchain. It maintains an expanding list of ”blocks,” or immutable records [1], [50], [51], [52], and [53]. Blocks are linked together using a hash produced by a cryptographic technique, as seen in Fig. 2 [54]. Because of this, blockchain can function as a trustworthy way to record transactions [55]. The peer-to-peer network (Fig. 3) ensures that all nodes have a copy of the full ledger and automatically corrects any node that attempts a fraudulent change. This leads to redundancy and security and eliminates the need for a central authority, [56].

Six levels make up the blockchain system, as seen in [57], and [58]. The core of blockchain architecture is the data layer, which includes time stamping, chain structures, and blocks. Blocks function as storage containers for transactions and their metadata in this layer. Bitcoin serves as an illustration of this, as each block comprises the following necessary components: the block size, block header, transaction counter, and transactions [55]. A cryptographic hash technique is applied to the block header in order to create a block hash, which is used to guarantee the unique identification of every block in the

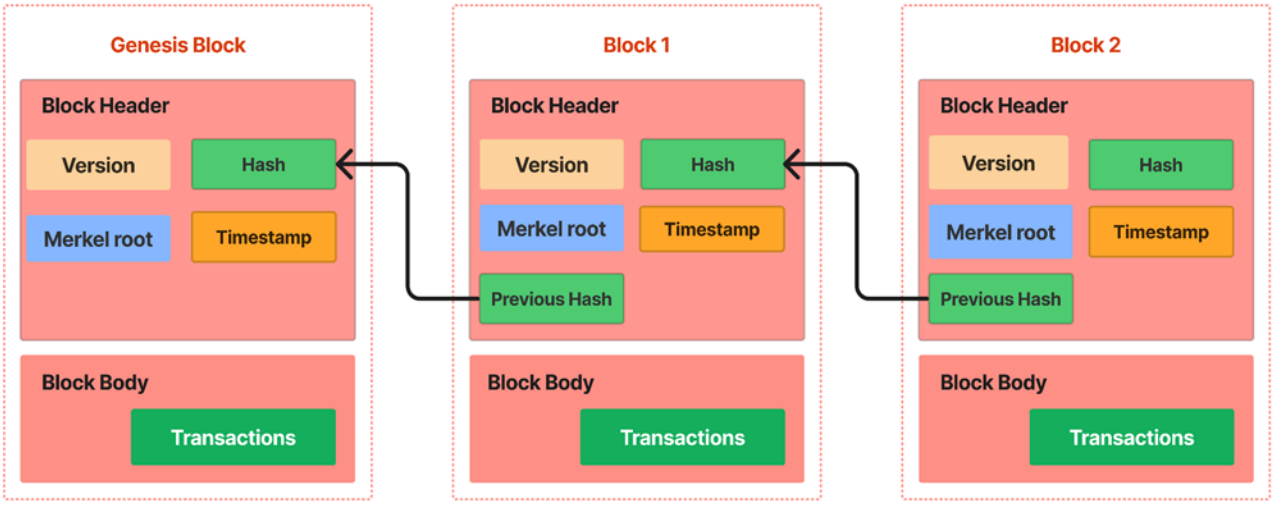


Fig. 2: Blockchain structure

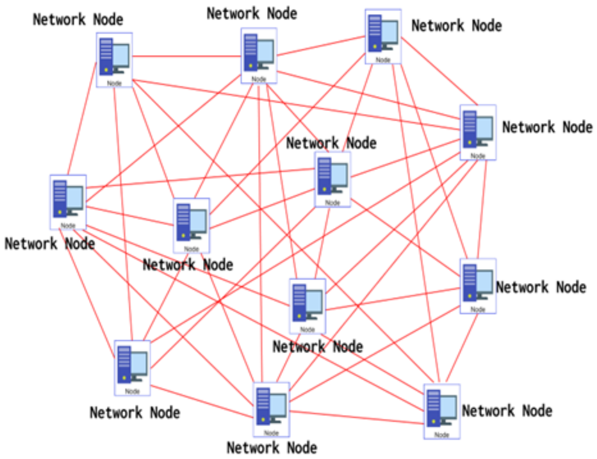


Fig. 3: Peer-to-Peer network

blockchain [1], [53], [59]. The SHA256 hash algorithm is used in the instance of Bitcoin. Additionally, the hash of the parent block is incorporated into the header of each block, forming a smooth and temporal connection between the blocks (Fig. 2). The genesis block is the first block in a blockchain that exists on its own without a parent block (Fig. 2). The interconnected structure created by the blockchain's construction ensures the security and integrity of the data stored on it. Its chain of information is immutable since every block has a distinct hash. The blockchain is extremely resistant to tampering since any attempt to change the content of a single block will result in an inconsistent chain. These characteristics offer a solid foundation for safe, decentralized data management systems [59].

3.2.2 Consensus mechanisms

Blockchain's consensus mechanism, which ensures network participants agree on the veracity of transactions, is one of its key features. To maintain trust and reduce malicious activity within the network, a number of consensus algorithms, including popular ones like Proof of Work (POW), Proof of Stake (POS), and Practical Byzantine Fault Tolerance (PBFT), have been developed. The popular consensus classification is displayed in Fig. 4. Blockchain networks attain high-security levels and do away with the possibility of a single point of failure through consensus.

- Proof of Work (PoW): Stands as a pioneering consensus mechanism within the realm of blockchain technology, [60]. To create new blocks for the blockchain, it uses the computational competition principle. In Proof of Work (PoW), miners perform calculations to yield a value, and the winner is the miner who is able to create a value that is less than the network's predetermined threshold [59], [60]. Proposals have been made for Proof of Weight, Proof of Reputation, Proof of Space, Proof of History, and Proof of Burn as variations of Proof of Work.

- Proof-of-Stake (PoS): This method has a major benefit over Proof-of-Work (PoW) in that it does not require expensive mining equipment [43]. Nodes have the option to mine or validate blocks in a Proof of Stake (PoS) system according to their stake. The latter is simply the quantity of coins they possess [60], [61]. With this method, users buy cryptocurrency and use it to get access to block creation opportunities. Introduced in [61], Delegated Proof-of-Stake (DPoS) is an

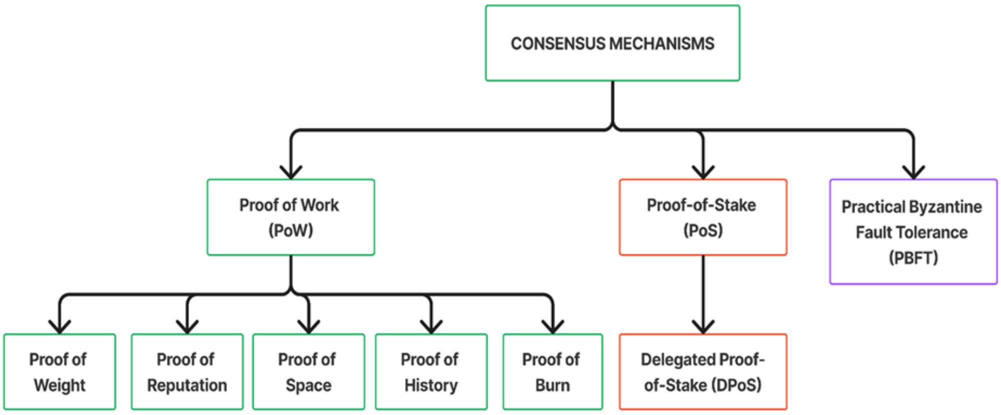


Fig. 4: Classification of Blockchain Consensus Mechanism

additional variant of Proof of Stake.

- Practical Byzantine Fault Tolerance (PBFT): is a consensus algorithm to ensure fault tolerance in distributed systems, especially when there are malicious or faulty nodes [62].

Compared to Proof-of-Work and Proof-of-Stake, PBFT does not depend on stake-based or mining processes. Rather, to reach a consensus, PBFT makes use of a sequence of message exchanges between nodes. A designated leader node in PBFT proposes a block of transactions to start the consensus process. Other nodes participate in a multi-round voting procedure following a node's proposal of a block. They converse with one another during this procedure to ascertain the validity of the suggested block [62]. When a significant number of nodes come to an agreement, the suggested block is put into the blockchain and deemed approved. By using PBFT, the system may withstand a specific number of malfunctioning nodes and still retain its liveness and safety characteristics [62].

3.2.3 Smart contracts

A smart contract is a computer protocol designed to autonomously execute, enforce, verify, and restrict the execution of its instructions. It makes it possible to execute transactions without the use of middlemen between anonymous or untrusted parties [63]. These are irreversible and traceable transactions. The components of a smart contract include value, address, function, and state. The associated code is run when a transaction is input; this results in an output event and a state change that is determined by the functional logic that has been defined. Every party to the smart contract agrees in advance to its terms and conditions, including its triggering scenarios, state transition procedures, and liabilities for breaking the contract. The smart contract is then deployed on the blockchain as code, and it will start working automatically as soon as the predefined requirements are satisfied. Ethereum is the most widely used platform for the development of smart contracts. According to [64], Ethereum outperforms Hyperledger Fabric in terms of the quantity of transactions that are completed successfully. The majority of developers write smart contracts using Solidity and Serpent. Chain code, also known as smart contracts, can be implemented using Hyperledger Fabric. Usually, Go or Java is used to develop it; the source [64] states that Go is the preferred language for best performance.

3.2.4 Oracles

The difficulty of blockchain technology to directly access external data initially hampered its integration with the real world. The idea of oracles was presented as a way around this restriction. In order to connect the blockchain to other domains and facilitate its use across multiple industries, oracles serve as centralized, reliable third parties that supply blockchain with real-world data (Fig. 5). Consensus oracles and centralized oracles are two different kinds. Consensus oracles, in contrast to centralized oracles, involve groups of oracles and are managed by a single authority [65].

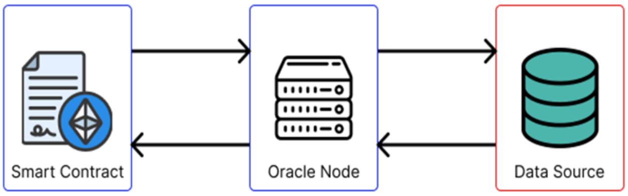


Fig. 5: An example of how Oracle interacts with smart contracts

3.3 Overview of IPFS

The Interplanetary File System (IPFS) is a decentralized file storage system that connects all computing devices to a single file system. To better understand IPFS, we can compare it to the Internet, but instead of depending on centralized servers, IPFS works using the collective power of the computers connected to the network [66]. In practice, IPFS works a bit like BitTorrent [60], where files are shared among users in a decentralized manner. However, IPFS goes a step further by using a hash system to uniquely identify files. Each file is assigned a unique ”hash” based on its content, called a Content Identifier (CID). This means that if a file's content changes, its hash changes as well, ensuring that files are intact and unaltered. This CID consists of 4 fields [67]:

- Multibase prefix: which indicates one of the 24 basic encoding methods used to create the binary content identifier (CID).
- CID version identifier: which indicates the version of the CID. There are currently two versions (v0 and v1).
- Multicode identifier: indicating how the addressed data was encoded.
- Multihash: which contains metadata indicating the default hash function used (SHA-256) and the default length (32 octets) of the actual hash of the contents. The term ”multihash” comes from the fact that it can support any hash algorithm.

CID = <**Multibase**> (*cid-version* || *multicode* || *multihash*)

When content is added to IPFS, it is divided into chunks (256kb by default), and each chunk is given its own CID (Content Identifier). The CID of each chunk is obtained by applying a hash function to its content and adding the metadata mentioned above. Once all chunks have a CID, IPFS constructs a Merkle Directed Acyclic Graph (Merkle DAG) of the file [66], [68]. This DAG is the form in which the file is provided by the original content editor. A Merkle DAG is a data structure similar to a Merkle tree but without balance requirements. The root node combines all the CIDs of its descendant nodes to form the final content CID (commonly referred to as the root CID). In addition, all files exchanged on the IPFS network are stored in Distributed Hash Tables (DHTs) [11], [66], [67]. A DHT is essentially a distributed data structure that allows information to be stored and retrieved using keys (in this case, CIDs) to obtain the corresponding values (PeerIDs and associated location information); the IPFS DHT is based on the Kademlia protocol [66], [69], which is a well-established technology for managing distributed DHTs and is similar to the way the BitTorrent Mainline DHT [70] works for file distribution on the BitTorrent network. It's worth noting that one of the advantages of IPFS is that it has no single point of failure. This means that there is no central server on which the entire network rests, making it more robust and resilient. IPFS network nodes work together to store and share files, and they don't need to trust each other for the system to work.

4 Related Works

In this section, we review recent research related to the blockchain-based verification of academic documents. Among the proposed approaches, a few caught our attention: The study in [12] outlines the importance of certificate verification and its impact on our society. It briefly discusses traditional verification methods and their limitations and proposes a blockchain-based graduation certificate verification system. This system is used not only for verification but also for generating new certificates; it generates digitally signed certificates using the asymmetric key and timestamp. Students receive a copy to use as they wish, and employers can verify the authenticity of these documents through the system by entering the public key of the university (issuing institution) and the digital signature applied. The authors in [13] discuss the current verification process and the proliferation of fake credentials. They proposed, as a proof-of-concept, a prototype of an open-source blockchain-based Ark platform that aims to provide higher education institutions with a credit and rating system and potential employers with a tool to validate a candidate's academic information. However, the article did not clearly present the technical details of implementing the proposed solution. The work in [14] proposed a solution that involves creating a platform for all the credentials that a student may possess. Through the platform, students store all their diplomas on the blockchain. To verify a diploma, a person needs the student's login and password. The consensus algorithm used for validation is Proof of Work (PoW), but some details regarding validation are not clearly disclosed. The study in [11] offers an interesting implementation of a new document verification system that combines blockchain and the InterPlanetary File System (IPFS) to increase the efficiency of detecting a forged document. However, he points out that one of the limitations of this system is that it only checks the availability of documents, but not their integrity, i.e., their content; in other words, the system checks changes made to the file without examining the content of the file. As an alternative, he suggests implementing Optical Character Recognition (OCR) in the system to overcome the limitation of verifying file content, and to create a better document verification system based on blockchain technology. Instead of the suggestion proposed in [11] to use OCR, we outline below our approach to effectively solve this major limitation of current blockchain-based document verification systems.

5 Solution Overview

Our proof of concept implements basic features to demonstrate how LLM is integrated to enhance security and avoid malicious certification actions. The system has 3 main interactions:

- **Document Request:** The student requests his academic document from his institution and awaits the request to be processed (step 1 in Fig. 6). Once ready he receives an email with the link to the requested document that he can download and share with other institutions (step 2* in Fig. 6).
- **Certification process:** The institution will proceed with its normal process to check student information and the validity of the requested document. Once the document is ready, an authorized staff member will use our platform for certification and send the document to the student by mail (steps 2, 2', 2" and 2"' in Fig. 6).
- **Verification by other institutions:** A third-party institution uses a mobile application by scanning the QR code, using the link added at the bottom of the document (Fig. 7), or inserting the document hash to check the authenticity of the document (step 3 in Fig. 6).

The main contribution of this paper resides in steps 2' and 2" in Fig. 6. We added LLM to extract key information that makes a document unique. Table 2 indicates key information for a student transcript or academic diploma based on the student transcript sample in Fig. 7. Therefore, before saving the document in blockchain or IPFS, the LLM extracts key information and checks in the blockchain if the hash gotten by hashing this set of information exist. It should be noted that the IPFS here is used to store the uploaded file in the decentralized network and the blockchain stores the key information hash and the IPFS file hash.

5.1 System Flow

The three main processes as stated in the previous section are the Document Request process, Certification process, and Verification process. The request process is the simplest one. The student only has to provide information such as his personal information, his academic information, and the documents he needs. The certification process is described in Fig. 8. The authorized staff member first login before taking further actions. For certification purposes, he will provide the PDF of the document. The content of the document is then concatenated with the prompt provided in Fig. 9. After the LLM has returned the desired response, it is hashed with the sha256 algorithm and used to check if such hash already exists in the blockchain. We save this file in the IPFS network only if the hash does not exist. And at last, save the IPFS file hash, document hash, and timestamp in the blockchain.

Table 2. Key information that makes a student transcript unique

Key
Document Title
Student Name
Year of Study
Option
Academic Year

The Verification process remains simple and is described in Fig. 10. The third-party institution may use the URL provided at the bottom of the document (Fig. 7) or scan the QR code to get the document hash. Given the hash, the platform returns the real file URL saved in the IPFS network corresponding to the hash provided, or else it returns an Error message.

5.2 Prototype Implementation

In a more detailed view, our system has connected 4 technologies (Fig. 11). The smart contract is written in Solidity, the web application developed in Python and accessible via http://127.0.0.1:5000 (Fig. 12), the IPFS network (locally installed), OpenAI as LLM and Ethereum network simulated by Ganache. Details on the development environment are described in Table 3.

The web application is divided into three sections: *An administration panel, a Request Page, and a Verification page for third-party institutions.*

Table 3. Development Environment

Component	Description
Hardware	Intel(R) Core (TM) i7-4600M CPU @ 2.90GHz 2.90 GHz
Memory	12.0 Go
Operating System	Windows 10 Professional
Blockchain Platform	Ethereum with Ganache
IPFS Network	Local – Desktop version
Programming Language	Python, Solidity

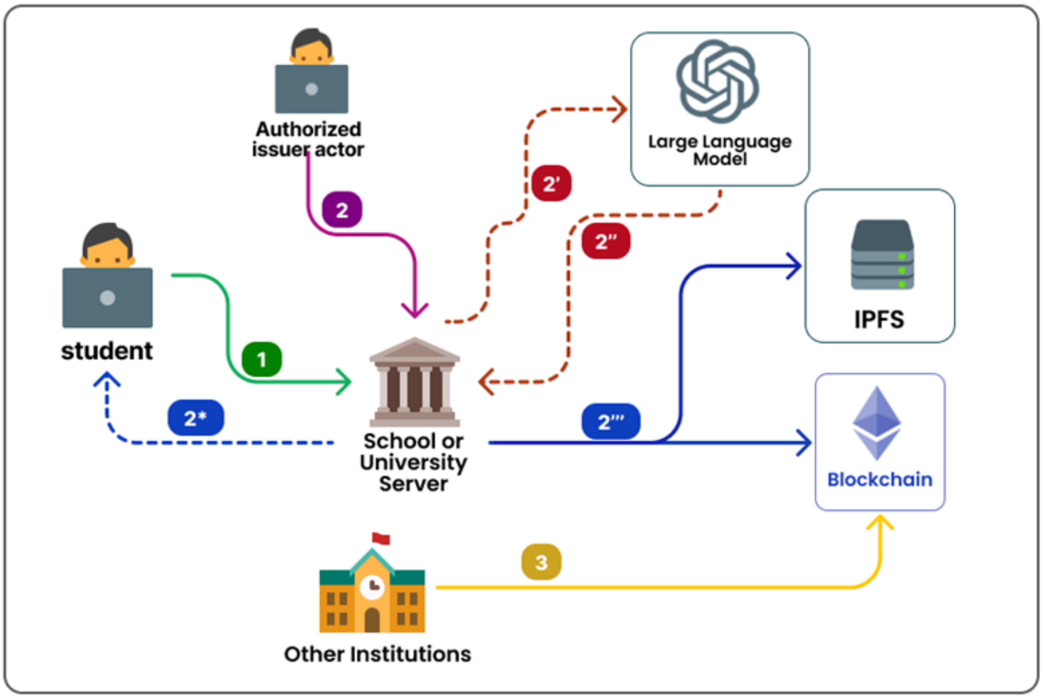
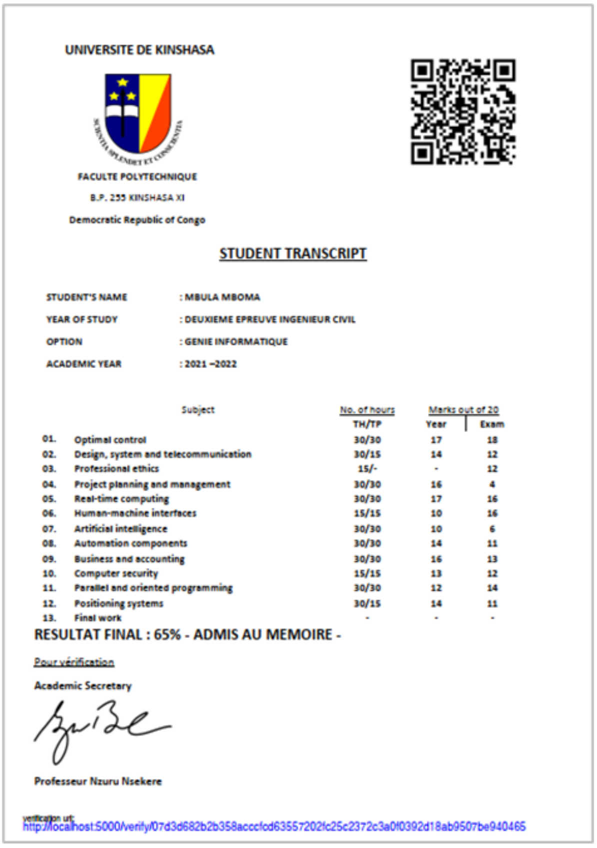


Fig. 6: System actors and Operation flow



UNIVERSITE DE KINSHASA

FACULTE POLYTECHNIQUE
B.P. 255 KINSHASA XI
Democratic Republic of Congo

STUDENT TRANSCRIPT

STUDENT'S NAME	: MBULA MBOMA
YEAR OF STUDY	: DEUXIEME EPREUVE INGENIEUR CIVIL
OPTION	: GENIE INFORMATIQUE
ACADEMIC YEAR	: 2021 –2022

	Subject	No. of hours	Marks out of 20	
		TH/TP	Year	Exam
01.	Optimal control	30/30	17	18
02.	Design, system and telecommunication	30/15	14	12
03.	Professional ethics	15/-	-	12
04.	Project planning and management	30/30	16	4
05.	Real-time computing	30/30	17	16
06.	Human-machine interfaces	15/15	10	16
07.	Artificial intelligence	30/30	10	6
08.	Automation components	30/30	14	11
09.	Business and accounting	30/30	16	13
10.	Computer security	15/15	13	12
11.	Parallel and oriented programming	30/30	12	14
12.	Positioning systems	30/15	14	11
13.	Final work	-	-	-

RESULTAT FINAL : 65% - ADMIS AU MEMOIRE -

Pour vérification

Academic Secretary

Professeur Nzuru Nsekere

verification url:
http://localhost:5000/verify/07d3d682b2b358acccfcd63557202fc25c2372c3a0f0392d18ab9507be940465

Fig. 7: Sample of a certified student transcript

a. The Administration Panel: To access the administration zone, the authorized staff member has to successfully log in to the system. He can view requests and respond to a request by uploading the requested pdf file.

b. Request page and Verification page: These two pages are open to the public to allow students to request documents or other institutions to verify them.

c. OpenAI: The process of text extraction is excellently done by the general-purpose model GPT-3.5-Turbo-16k. So, we do not need to finetune a model for this purpose. He receives the prompt (Fig. 10) and the file content and returns the key information as requested in the prompt..

d. IPFS network: it is a peer-to-peer content delivery network that stores stores, retrieves, and locates data based on the fingerprint of its actual content rather than its name or location. For our solution, we install a single node in our computer for test purposes. The backend connects to the IPFS via an API with http://localhost:5001/api/v0 as the base URL to push files. The URL of the certified file sent to the user has this format: http://127.0.0.1:8080/ipfs/filehash.

e. Ethereum blockchain: To avoid using the public Ethereum network, we used Ganache as a truf-

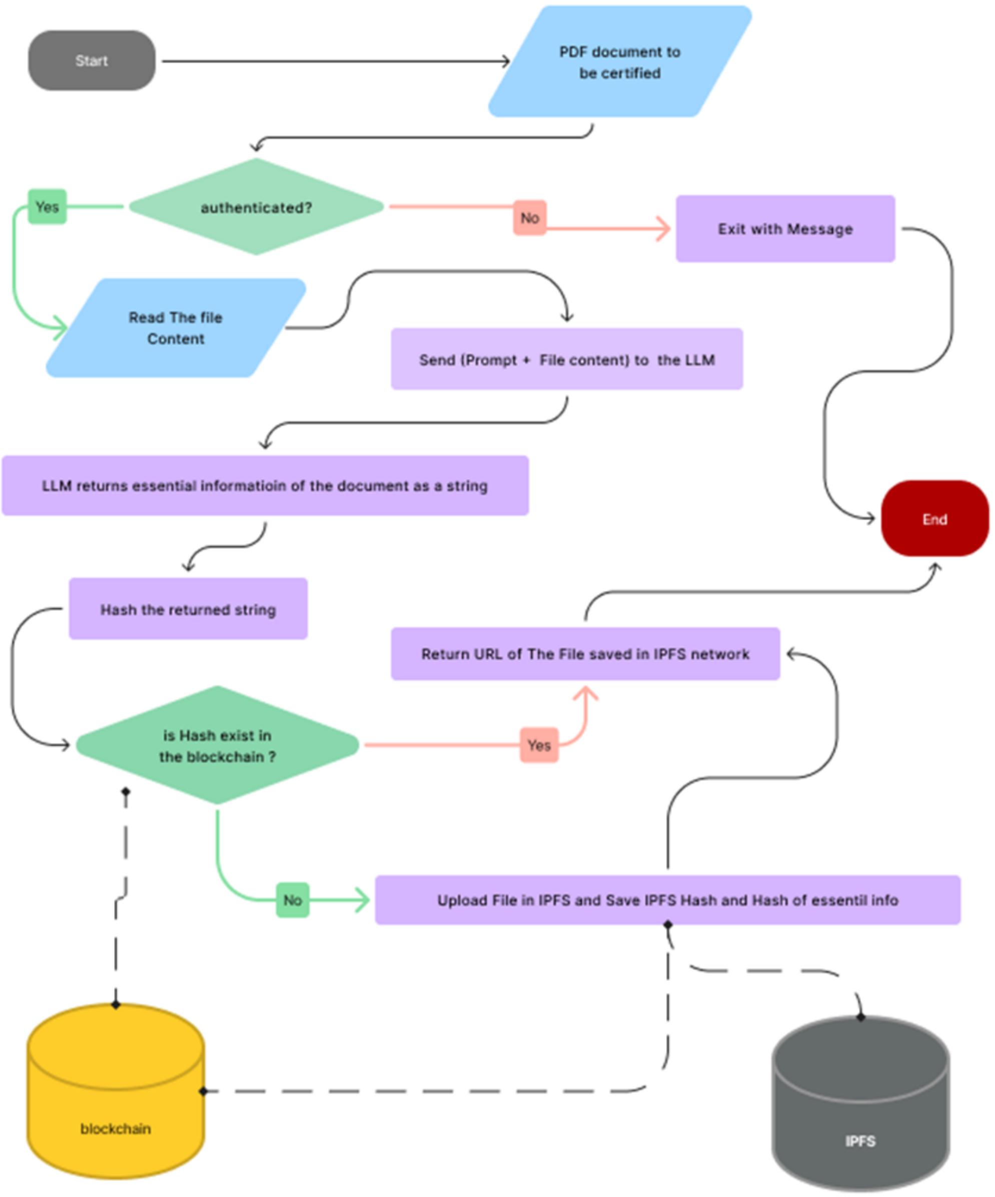

Start
PDF document to be certified
authenticated?
Yes
No
Exit with Message
Read The file Content
Send (Prompt + File content) to the LLM
LLM returns essential informatioin of the document as a string
End
Hash the returned string
Return URL of The File saved in IPFS network
is Hash exist in the blockchain ?
Yes
No
Upload File in IPFS and Save IPFS Hash and Hash of essentil info
blockchain
IPFS

Fig. 8: Flow diagram for the Certification Process

fle Suite to have a test network. The backend connects to the smart contract through the RPC endpoint http://127.0.0.1:7545.

5.3 Results and Discussion

Our prototype implementation successfully demonstrated that the integration of LLM added a layer of trust to the certification process. This trust is established through the rigorous verification of the hash obtained after encrypting key information of the document with the sha256 algorithm. The test result shown in Table 4 demonstrated that if key information of a document, as described in Table 2, remains unchanged, the hash value remains the same, indicating that the uploaded file may be considered fraudulent. Conversely, if any key information, as described in Table 2, is modified or missing, it results in a change of the hash value, making the document different or not yet certified. Our solution is flexible and no huge modification is required if the type of the document is changed. The solution may be used in any other industry because only key information will change. Thus, the prompt will change or a finetuning will be required based on the complexity of the document.

Integrating LLM into the certification process raises several challenges. Privacy and data security concerns are at the top of the list. The Issuing Institutions must ensure that sensitive student data is handled securely and in compliance with relevant regu-

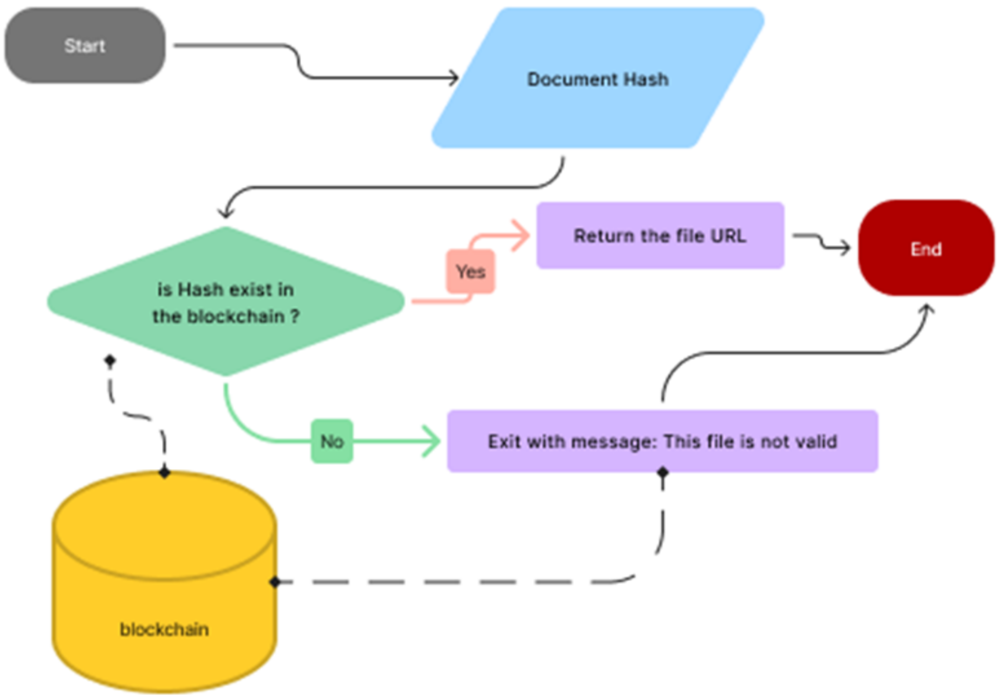

Start
Document Hash
is Hash exist in the blockchain ?
Yes
Return the file URL
End
No
Exit with message: This file is not valid
blockchain

Fig. 9: Flow diagram for the Verification Process

Return only a JSON with the following attributes from the transcript document below: `"document_title," "student_name," "academic_year," "option," and "year_of_study."`. Please note that no additional text should be included except the JSON.
NOTE: DO NOT PROVIDE ANY OTHER TEXT EXCEPT THE VALID JSON, AND THE EXTRACTED INFO HAS TO BE IN LOWERCASE

Fig. 10: The Prompt used to extract key information from the document. This prompt is concatenated with the document content and sent to the LLM

Table 4. Summary of Use cases and results obtained by changing document content and evaluating hash.

Case	HASH	Remark
key information of a document is not modified	Remains unchanged	If key information as described in Table 2 is not modified and any other info is modified the Hash value will remain the same and the uploaded file will be considered fraudulent
Missing key information in the document	Changes	In this case, the document is considered different and the student cannot use a document with missing information
Any of the key information is changed	Changes	In this case, the student may be requesting a transcript of another academic year for example. Thus, we consider that it is a different document

lations. To enforce security, we suggest that the platform be fully decentralized to avoid human interactions with the centralized server. It means that the backend should be replaced only by the smart contract for the entire work. Another challenge resides in the accuracy of LLM in extracting key information. In this work, we use the general-purpose model because it performs accurately. But in the case where the document is too complex and requires a fine-tuned model. In this situation, the institution should contin-

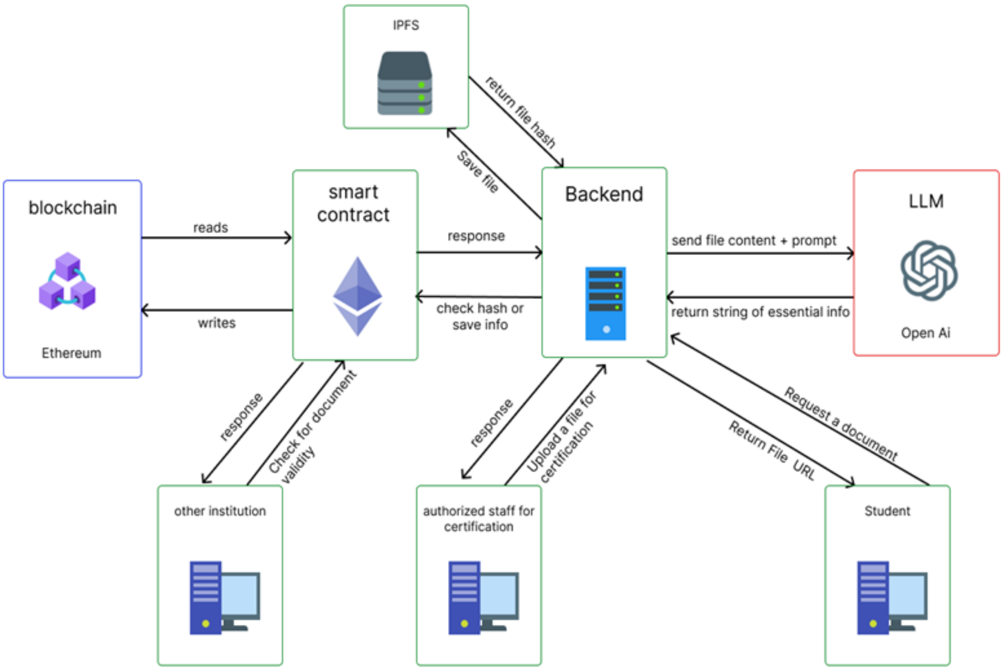
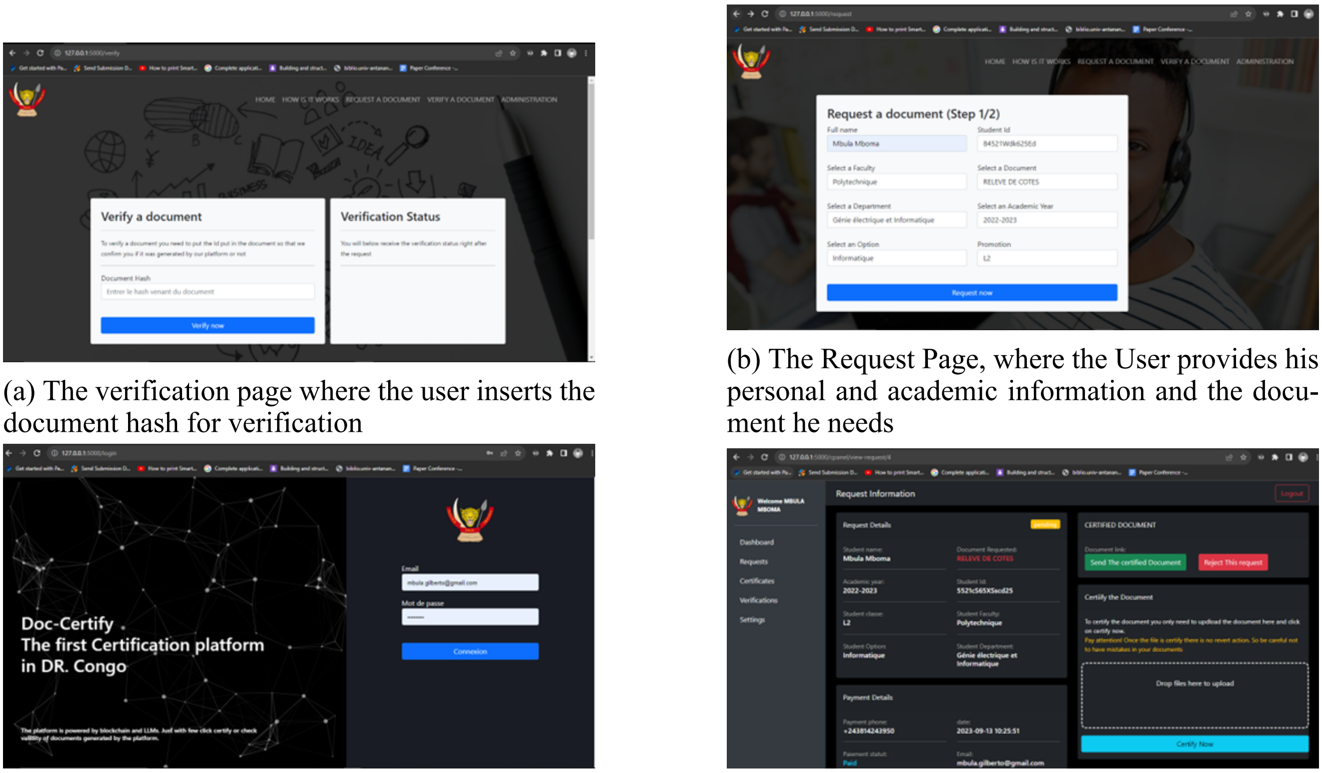


Fig. 11: Architecture of the entire system. The backend sends the document content to the LLM so that this one returns back the key information of the document to be hashed. And the other institutions verify the integrity of documents via the Blockchain


127.0.0.1:5000/verify
HOME HOW IS IT WORKS REQUEST A DOCUMENT VERIFY A DOCUMENT ADMINISTRATION

Verify a document

To verify a document you need to put the Id put in the document so that we confirm you if it was generated by our platform or not

Document Hash

Entrer le hash venant du document

Verify now

Verification Status

You will below receive the verification status right after the request

(a) The verification page where the user inserts the document hash for verification

127.0.0.1:5000/request
HOME HOW IS IT WORKS REQUEST A DOCUMENT VERIFY A DOCUMENT ADMINISTRATION

Request a document (Step 1/2)

Full name: Mbula Mboma

Student Id: 84521Wdk625Ed

Select a Faculty: Polytechnique

Select a Document: RELEVE DE COTES

Select a Department: Génie électrique et Informatique

Select an Academic Year: 2022-2023

Select an Option: Informatique

Promotion: L2

Request now

(b) The Request Page, where the User provides his personal and academic information and the document he needs

Doc-Certify .
The first Certification platform
in DR. Congo

The platform is powered by blockchain and LLMs. Just with few click certify or check valitity of documents generated by the platform.

Email: mbula.gilberto@gmail.com

Mot de passe: ••••••••

Connexion

(c) The login page where the authorized Staff member puts his email and password for authentication

Request Information

Request Details pending

Student name: Mbula Mboma

Document Requested: RELEVE DE COTES

Academic year: 2022-2023

Student Id: 5521c565XSacd25

Student classe: L2

Student Faculty: Polytechnique

Student Option: Informatique

Student Department: Génie électrique et Informatique

Payment Details

Payment phone: +243814243950

date: 2023-09-13 10:25:51

Paiement statut: Paid

Email: mbula.gilberto@gmail.com

CERTIFIED DOCUMENT

Document link:

Send The certified Document | Reject This request

Certify the Document

To certify the document you only need to updload the document here and click on certify now.

Pay attention! Once the file is certify there is no revert action. So be careful not to have mistakes in your documents

Drop files here to upload

Certify Now

(d) The Certification page. The Administrator only has to drag and drop the file to be certified

Fig. 12: Main Pages of the Web application platform

uously evaluate and fine-tune the model to ensure accurate results. Finally, the integration of LLM in the certification process adds more latency in the request because the communication with the LLM mostly depends on the network bandwidth.

6 Conclusion

In this article, we present a new approach to improve the security of academic document verification systems. We are convinced that to achieve fast, easy, and credible verification, it is essential to take rigorous care of each document to be saved in our system; therefore, we have implemented an additional security mechanism during the certification and file upload to avoid any forgery or fraudulent maneuvers. This security mechanism is based on the use of GPT-3.5 to extract key information from the document and encrypt it using a hash function (SHA-256). This hash is then used for checking if any previously saved document has the same signature. This additional check ensures that any student transcript or diploma is issued twice with different content. We also have presented a hybrid architecture where a centralized server is served to issue documents, a decentralized blockchain to store file hash and document content hash, and an interplanetary file storage system to store issued documents. The system stores, generates, and delivers certified academic documents, but also verifies their authenticity on demand. The verification process is adapted to both digital and physical (printed) documents. By evaluating the impact of the integration of LLM in the certification process, we conclude that the run time increased by 40% and tightly depends on the network bandwidth. In terms of security, the certification institution must ensure that the LLM platform aligns with internal and government compliances in terms of data security and privacy.

As future research, our study can focus on optimizing algorithms to reduce run time and a deep study on security vulnerability that such integration comes up with. We also plan to modify our system to accommodate a wider variety of academic texts and languages. This development will involve tailoring the AI model to suit diverse document structures and language nuances, depending on the specific context. The source code used for this work is available on GitHub [71].

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors partially used the tool Grammarly to polish the grammar and some parts of the wording style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, “Toolformer: Language models can teach themselves to use tools,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 539–68 551, 2024.

[3] M. M. Rahman, M. T. K. Tonmoy, S. R. Shihab, and R. Farhana, “Blockchain-based certificate authentication system with enabling correction,” *arXiv preprint arXiv:2302.03877*, 2023, https://doi.org/10.48550/arXiv.2302.03877.

[4] M. Gonzalez-Lee, C. Santiago-Avila, M. Nakano-Miyatake, and H. Perez-Meana, “Watermarking based document authentication in script format,” in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*. Cancun, Mexico: IEEE, 2009, pp. 837–841,
https://doi.org/10.1109/MWSCAS.2009.5235898.

[5] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J.-M. Gaudin, and C. Guichard, “Two-level qr code for private message sharing and document authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 571–583, 2015,
https://doi.org/10.1109/TIFS.2015.2506546.

[6] A. T. Arief, W. Wirawan, and Y. K. Suprapto, “Authentication of printed document using quick response (qr) code,” in *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. Surabaya, Indonesia: IEEE, 2019, pp. 228–233,
https://doi.org/10.1109/ISITIA.2019.8937084.

[7] M. Salleh and T. C. Yew, “Application of 2d barcode in hardcopy document verification system,” in *Advances in Information Security and Assurance: Third International Conference and Workshops, ISA 2009, Seoul, Korea, June 25-27, 2009. Proceedings 3*. Seoul, Korea: Springer, 2009, pp. 644–651,
https://doi.org/10.1007/978-3-642-02617-1_65.

[8] A. Husain, M. Bakhtiari, and A. Zainal, "Printed document integrity verification using barcode," *Journal Teknologi (Sciences and Engineering)*, vol. 70, no. 3, pp. 99–106, 2014.

[9] C. M. Li, P. Hu, and W. C. Lau, "Authpaper: Protecting paper-based documents and credentials using authenticated 2d barcodes," in *2015 IEEE International Conference on Communications (ICC)*. London, UK: IEEE, 2015, pp. 7400–7406, https://doi.org/10.1109/ICC.2015.7249509.

[10] M. A. A. Alameri, B. Ciylan, and B. Mahmood, "Computational methods for forgery detection in printed official documents," in *2022 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*. Alexandria, Egypt: IEEE, 2022, pp. 307–313, https://doi.org/10.1109/ICETSIS55481.2022.9888875.

[11] M. D. R. Zainuddin and K. Y. Choo, "Design a document verification system based on blockchain technology," in *Multimedia University Engineering Conference (MECON 2022)*. Melaka, Malaysia: Atlantis Press, 2022, pp. 229–244, https://doi.org/10.2991/978-94-6463-082-4_23.

[12] O. Ghazali and O. S. Saleh, "A graduation certificate verification model via utilization of the blockchain technology," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 3-2, pp. 29–34, 2018.

[13] J. G. Dongre, S. M. Tikam, and V. B. Gharat, "Education degree fraud detection and student certificate verification using blockchain," *Int. J. Eng. Res. Technol*, vol. 9, no. 4, pp. 300–303, 2020.

[14] M. R. Suganthalakshmi, M. C. Praba, M. K. Abhirami, M. S. Puvaneswari, and A. Prof, "Blockchain based certificate validation system," 2022. [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_7_july_2022/28889/final/fin_irjmets1659003745.pdf

[15] S. Jayalakshmi and Y. Kalpana, "A private blockchain-based distributed ledger storage structure for enhancing data security of academic documents." *Grenze International Journal of Engineering & Technology (GIJET)*, vol. 9, no. 1, pp. 25–35, 2023.

[16] F. M. Enescu, N. Bizon, and V. M. Ionescu, "Blockchain technology protects diplomas against fraud," in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. Pitesti, Romania: IEEE, 2021, pp. 1–6, https://doi.org/10.1109/ECAI52376.2021.9515107.

[17] A. Gayathiri, J. Jayachitra, and S. Matilda, "Certificate validation using blockchain," in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*. Chennai, India: IEEE, 2020, pp. 1–4, https://doi.org/10.1109/ICSSS49621.2020.9201988.

[18] I. T. Imam, Y. Arafat, K. S. Alam, and S. A. Shahriyar, "Doc-block: A blockchain based authentication system for digital documents," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. Vellore, India: IEEE, 2021, pp. 1262–1267, https://doi.org/10.1109/ICICV50876.2021.9388428.

[19] N. Malsa, V. Vyas, J. Gautam, A. Ghosh, and R. N. Shaw, "Certbchain: a step by step approach towards building a blockchain based distributed application for certificate verification system," in *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*. Greater Noida, India: IEEE, 2021, pp. 800–806, https://doi.org/10.1109/ICCCA52192.2021.9666311.

[20] A. D. B. Machado, M. Sousa, and F. D. S. Pereira, "Applications of blockchain technology to education policy," *Applications of blockchain technology to education policy*, pp. 157–163, 2019.

[21] V. Yfantis and K. Ntalianis, "A blockchain platform for teaching services among the students," *WSEAS Transactions on Advances in Engineering Education*, vol. 19, pp. 141–146, 2022.

[22] M. Shanahan, "Talking about large language models," *Communications of the ACM*, vol. 67, no. 2, pp. 68–79, 2024, https://doi.org/10.1145/3624724.

[23] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan,

R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[24] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[25] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic, "Galactica: A large language model for science," *arXiv preprint arXiv:2211.09085*, 2022, https://doi.org/10.48550/arXiv.2211.09085.

[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023, https://doi.org/10.48550/arXiv.2302.13971.

[27] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024, https://doi.org/10.1145/3641289.

[28] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023, https://doi.org/10.48550/arXiv.2303.18223.

[29] S. M. Thede and M. Harper, "A second-order hidden markov model for part-of-speech tagging," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 1999, pp. 175–182.

[30] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "A tree-based statistical language model for natural language speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1001–1008, 1989, https://doi.org/10.1109/29.32278.

[31] T. Brants, A. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 858–867.

[32] X. Liu and W. B. Croft, "Statistical language modeling for information retrieval." *Annu. Rev. Inf. Sci. Technol.*, vol. 39, no. 1, pp. 1–31, 2005.

[33] C. Zhai, "Statistical language models for information retrieval a critical review," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 3, pp. 137–213, 2008. [Online]. Available: 10.1561/1500000008

[34] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013, https://doi.org/10.48550/arXiv.1301.3781.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[37] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017, https://doi.org/10.1162/tacl_a_00051.

[38] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.

[39] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR abs/1802.05365 arXiv preprint arXiv:1802.05365*, 2018, https://doi.org/10.48550/arXiv.1802.05365.

[40] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020, https://doi.org/10.48550/arXiv.2001.08361.

[41] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An explanation of in-context learning as implicit bayesian inference," *arXiv preprint arXiv:2111.02080*, 2021, https://doi.org/10.48550/arXiv.2111.02080.

[42] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021, https://doi.org/10.48550/arXiv.2109.01652.

[43] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[44] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022, https://doi.org/10.48550/arXiv.2203.15556.

[45] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3505–3506, https://doi.org/10.1145/3394486.3406703.

[46] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019, https://doi.org/10.48550/arXiv.1909.08053.

[47] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[48] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[49] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021, https://doi.org/10.48550/arXiv.2112.09332.

[50] R. T. Kwok and B. Maurice, "Aperiodic linear complexities of de bruijn sequences," in *Advances in Cryptology—CRYPTO'88: Proceedings 8*. Springer, 1990, pp. 479–482, https://doi.org/10.1007/0-387-34799-2_33.

[51] S. Haber and W. S. Stornetta, "Secure names for bit-strings," in *Proceedings of the 4th ACM Conference on Computer and Communications Security*, 1997, pp. 28–35.

[52] J. G. M. Mboma, O. T. Tshipata, W. V. Kambale, and K. Kyamakya, "Assessing how large language models can be integrated with or used for blockchain technology: Overview and illustrative case study," in *2023 27th International Conference on Circuits, Systems, Communications and Computers (CSCC)*. Rhodes (Rodos) Island, Greece: IEEE, 2023, pp. 59–70, https://doi.org/10.1109/CSCC58962.2023.00018.

[53] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[54] G. Fox, "Peer-to-peer networks," *Computing in Science & Engineering*, vol. 3, no. 3, pp. 75–77, 2001, https://doi.org/10.1109/5992.919270.

[55] M. Swan, *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc.", 2015.

[56] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564, https://doi.org/10.1109/BigDataCongress.2017.85.

[57] M. N. M. Bhutta, A. A. Khwaja, A. Nadeem, H. F. Ahmad, M. K. Khan, M. A. Hanif, H. Song, M. Alshamari, and Y. Cao, "A survey on blockchain technology: Evolution, architecture and security," *Ieee Access*, vol. 9, pp. 61 048–61 073, 2021, https://doi.org/10.1109/ACCESS.2021.3072849.

[58] C. V. B. Murthy, M. L. Shri, S. Kadry, and S. Lim, "Blockchain based cloud computing: Architecture and research challenges," *IEEE access*, vol. 8, pp. 205 190–205 205, 2020, https://doi.org/10.1109/ACCESS.2020.3036812.

[59] S. Ghimire and H. Selvaraj, "A survey on bitcoin cryptocurrency and its mining," in *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE, 2018, pp. 1–6, https://doi.org/10.1109/ICSENG.2018.8638208.

[60] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," *arXiv preprint arXiv:2001.07091*, 2020, https://doi.org/10.48550/arXiv.2001.07091.

[61] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15, https://doi.org/10.1145/3190508.3190538.

[62] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[63] M. Alharby and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint arXiv:1710.06372*, 2017, https://doi.org/10.48550/arXiv.1710.06372.

[64] M. Dabbagh, M. Kakavand, M. Tahir, and A. Amphawan, "Performance analysis of blockchain platforms: Empirical evaluation of hyperledger fabric and ethereum," in *2020 IEEE 2nd International conference on artificial intelligence in engineering and technology (IICAIET)*. IEEE, 2020, pp. 1–6, https://doi.org/10.1109/IICAIET49801.2020.9257811.

[65] G. Caldarelli, "Overview of blockchain oracle research," *Future Internet*, vol. 14, no. 6, p. 175, 2022, https://doi.org/10.3390/fi14060175.

[66] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014, https://doi.org/10.48550/arXiv.1407.3561.

[67] D. Trautwein, A. Raman, G. Tyson, I. Castro, W. Scott, M. Schubotz, B. Gipp, and Y. Psaras, "Design and evaluation of ipfs: a storage layer for the decentralized web," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 739–752, https://doi.org/10.1145/3544216.3544232.

[68] C. Helbling, "Directed graph hashing," *arXiv preprint arXiv:2002.06653*, 2020, https://doi.org/10.48550/arXiv.2002.06653.

[69] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 53–65, https://doi.org/10.1007/3-540-45748-8_5.

[70] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," *arXiv preprint arXiv:2001.07091*, 2020, https://doi.org/10.48550/arXiv.2001.07091.

[71] M. Mboma, "document-certification-with-blockchain-authority-dashboard," 2023, accessed: Sep. 16, 2023. [Online]. Available: https://t.ly/w0Uj7

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)
The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.