



**HAL**  
open science

# Smart Contracts Implementation Based on Bidirectional Encoder Representations from Transformers

Bajeela Aejas, Abdelaziz Bouras, Abdelhak Belhi, Housseem Gasmi

► **To cite this version:**

Bajeela Aejas, Abdelaziz Bouras, Abdelhak Belhi, Housseem Gasmi. Smart Contracts Implementation Based on Bidirectional Encoder Representations from Transformers. 18th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2021, Curitiba, Brazil. pp.293-304, 10.1007/978-3-030-94335-6\_21 . hal-04186149

**HAL Id: hal-04186149**

**<https://inria.hal.science/hal-04186149v1>**

Submitted on 23 Aug 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Smart Contracts Implementation based on Bidirectional Encoder Representations from Transformers

Bajeela Aejas<sup>[1]</sup>[0000-0001-7717-6619], Abdelaziz Bouras<sup>[1]</sup>[0000-0001-5765-1259], Abdelhak Be-

lhi<sup>1</sup>, Housseem Gasmi<sup>1</sup>

<sup>1</sup>Qatar University, Doha, Qatar

{ba1901053, abdelaziz.bouras, abdelhak.belhi,  
housseem.gasmi}@qu.edu.qa

**Abstract.** The distribution and immutability properties of blockchains made it possible to use them in various fields, such as Supply Chain, finance and health. The automation of the creation and execution of transactions in a blockchain in a decentralized and transparent manner is realized through Smart Contracts programming codes. This paper presents the implementation of Smart Contracts in specific manufacturing Supply Chains and discusses their life cycle and impact on the Supply Chain management. The presented application deals with the possibility of transforming natural language contracts of a given Supply Chain to automated Smart Contracts that makes the Supply Chain management faster and safer. A first solution is proposed based on Bidirectional Encoder Representations from Transformers (BERT) model and limited to the implementation of Smart Contracts of the Supply Chain legal contracts. Also described here is the ways of extracting contract elements from legal contracts by applying the BERT Deep Learning method on annotated contract dataset of a corpus of 13000 annotations over 510 contracts.

**Keywords:** Blockchain, Smart Contract, Supply Chain, BERT, NLP.

## 1 Introduction

Blockchain has become one of the emerging research fields after the success of the first crypto currency, the Bitcoin<sup>1</sup> in 2009. After a decade, blockchain is now in its voyage in various fields in addition to cryptocurrencies such as insurance [1], Health [2], Supply Chain [3][4], Internet of Things [5], Building Information Modeling [6] etc. It can be defined as a shared and immutable ledger for transaction and control of goods and services of the Supply Chain. The main attraction of blockchain is its ability to automate the secure, transparent, and flexible business transactions.

Smart Contracts are the backbone of blockchain that control the transaction with rules that are to be satisfied for proceeding the transactions. They are the pieces of software that has written rules to be executed automatically to update the state of the

---

<sup>1</sup> <https://bitcoin.org/bitcoin.pdf>

block chain in a systematic way. They are powerful tools for creating decentralized applications without any third-party support.

This paper discusses the concepts of Smart Contracts and the use of Smart Contracts in manufacturing Supply Chain. Such Supply Chain involves the management of lots of legal contracts in various steps and between all the partners. The handling and management of such numerous contracts manually over the lifecycle of the products is in general tedious and time consuming. The automation of such contracts and their handling is a necessity nowadays and their conversion into Smart Contracts is one of the possible solutions in a decentralized and secure context due to the cryptographic handling within their supporting blockchain. The paper addresses two parts. First, the ongoing automation of entity extraction (clauses identification) from supply chain related legal contracts. This step needs the creation of annotated contract dataset and implementation of an appropriate NLP technique for entity recognition and extraction. The main challenge in this field is the unavailability of annotated contract dataset, especially in the supply chain domain. Since contracts are considered as highly confidential, for security reasons, most of the law firms and contract management companies do not make the dataset public. To create a dataset, a large number of contracts need to be manually annotated. This is very tedious and time-consuming work that needs thorough knowledge of the contract structure and content. A new annotated contract dataset called Contract Understanding Atticus Dataset (CUAD), is made public for the first time by a contract management project called the Atticus Project<sup>2</sup>. This dataset contains all type of legal contracts and provides a good input start for our research. This dataset is prepared as Question Answering task of NLP. The proposed model is implemented using Question Answering model of Hugging Face Transformers library [16]. Bidirectional Encoder Representations from Transformers (BERT) model is used for identifying the underlying clauses. The task is being experimented using python as a programming platform. The second part of the research is the automation of smart contract creation using the extracted clauses. This research will be a promising enhancement in the domain of blockchain to supply chain in terms of money, time, and effort.

Globally, the main issues addressed in this research are:

- Creation of supply chain related annotated dataset for this research. This includes selection of important contract clauses that is necessary for creation of an appropriate Smart Contracts.
- Selection of best NLP techniques to be used for extraction of these clauses.
- How to automate the creation of Smart Contract from the extracted clauses with efficiency.

## 2 Lifecycle of Smart Contracts

Smart Contracts are the scripts in a blockchain which help the automation of trading and transaction in a decentralized network. The idea of Smart Contract, first introduced by Nick Szabo [8], explains how to execute a contract securely between two parties

---

<sup>2</sup> <https://www.atticusprojectai.org/>

without the need of a third party. Smart Contracts have a simple if-then-else structure that is embedded in to the blockchain. The rules written in Smart Contracts are executed when the predefined conditions are verified and met. This can be anything like releasing of funds, issuing of tickets, registering a vehicle, sending notifications etc. Once the Smart Contract has been deployed, the transaction cannot be changed. Only the people participating in transaction can see and access the result. A Smart Contract has mainly three functions [9].

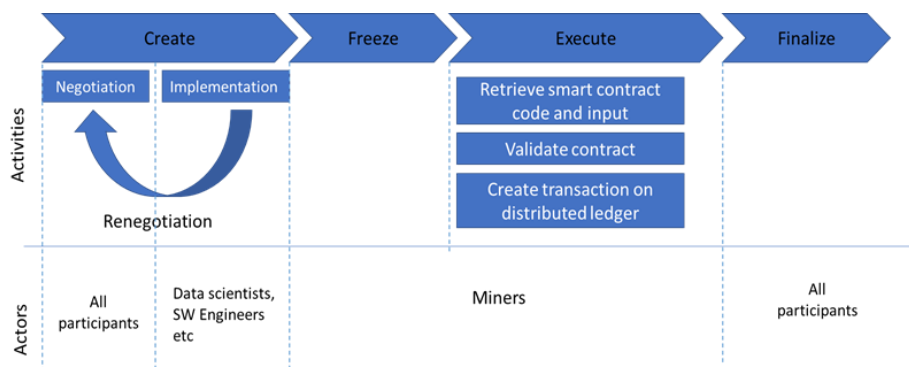
- Agreement between the parties. The contractual agreements between the parties are transformed to executable code. The transaction denotes the fulfillment of contractual obligation. The code evaluates the condition for fulfillment. This code is then stored in the blockchain.
- Precondition validation. Validation of whether the preconditions of the Smart Contract are met or not is done by the participating nodes.
- Execution of Smart Contract. If the preconditions are satisfied, the next step is the execution of Smart Contract. The participating nodes perform the transaction which is reflected throughout the blockchain.

## 2.1 Smart Contract- Life Cycle

The life cycle of Smart Contract is composed of the following phases[7] :

- Creation
- Deployment and Freezing
- Execution
- Finalization

These phases are shown in the Fig 1. hereunder:



**Fig. 1.** Life cycle of a Smart Contract [7]

**Creation of Smart Contract.** In this phase, parties involved in the transaction must agree on the terms and obligations of the contract. This negotiation phase is very similar to actual agreement negotiation. After the agreement on obligations, the contract must be turned in to code. This is the implementation phase. Implementation of the Smart Contract can be done in many high-level languages. One most used language is Solidity [10].

Now participants must agree to the coded version of the Smart Contract. After the agreement, the Smart Contract will be included in a distributed ledger. Thus it gets published in the blockchain. At this stage, all participants will receive the contract. After all nodes agree to this Smart Contract, it will start execution. In case any error occurs in the contract, reverting to previous state is not possible as Smart Contracts are decentralized. So, when an error occurs, a new contract needs to be created.

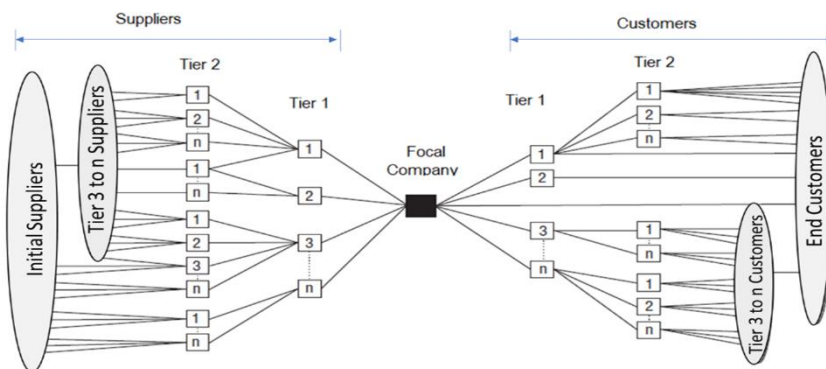
**Deployment and Freezing of Smart Contract.** After the submission of Smart Contracts to a blockchain, their validation depends on the majority of confirmations by the participants. At this freezing phase, any transaction made to the wallet addresses is frozen and nodes are in control of checking if the preconditions are met or not for validating the Smart Contract.

**Execution of Smart Contract.** After the preconditions are met, Smart Contract is now ready for execution by the nodes. During execution, many new transactions are added, and the current states are being updated throughout a distributed ledger and these are validated through consensus protocol.

**Finalization.** After the validation of transactions and states by consensus protocol, all the prior committed digital assets that were frozen, get transferred. And thus, the validated transactions confirm that the contract is fulfilled.

## 2.2 Contracts Automation for Supply Chain Management.

Supply Chain management is well known as a long and complex process. The Supply Chain includes all the activities from design and manufacturing through shipping to delivery of the products to customers. This may involve lots of supplier and customer chains [11]. As the structure becomes complex, the risk and efforts involved also become high. The complex structure of a Supply Chain is represented in Fig.2. Smart Contracts can contribute to simplify this complex process by improving visibility and tracking. The participants can decide and negotiate over the agreements and the decentralized nature of Smart Contract may help them to avoid dispute and track the smooth movements of goods and services with transparency.



**Fig. 2.** Supply Chain network structure [11]

Supply Chain management is one of the largest domains that deals with large number of legal contracts at a time. The life cycle of a Supply Chain product deals with management of different types of contracts. Contracts are legal documents signed by two or more parties that clearly explain the rights and duties of participants for the execution of any activities in the Supply Chain. The participants must be careful in following the clauses and activity durations defined in the contracts as this may lead to financial losses and disputes in future.

Usually, contract management is done manually by the legal authorities and staffs. This process is highly error prone, expensive and time consuming. Automation of contract management can solve many of the problems faced by current manual contract management system.

Since Supply Chain process involves handling a lot of contracts, the introduction of automated contract management in this domain can help a lot in speeding up the process and avoiding unwanted financial losses. Automation helps to get rid of the loopholes and manual mistakes that may arise during the manual contract management and thereby provides transparency on the contents.

The introduction of blockchain to the Supply Chain domain further increases the possibility of automation of contracts. The contract automation and conversion of contracts to Smart Contracts can help to make the contractual obligations to be handled more professionally in blockchain. Smart Contracts, as-discussed before, have a decentralized nature and are more secure. The idea of contract management through Smart Contracts for blockchain will make the Supply Chain management simpler and faster.

The main process in contract automation involves extraction of key entities from the contract. This can be done by performing Named Entity Recognition (NER) on the contract dataset. NER is a subset of Natural Language Processing (NLP) that deals with recognition of main entities from the given text [19].

There are various Machine Learning and Deep Learning methods for performing NER. Many pre-trained NER applications such as Stanford NER<sup>3</sup>, SpaCy<sup>4</sup> etc. are also available. The problem of using pre-trained NER models for contract element extraction is that these are trained to identify general named entities such as person, place, date etc. In the case of contracts, these have specific structure and usage. Hence, pre-trained NER models perform poorly for such domain specific dataset [18].

Domain specific NER needs the model to be trained in domain specific datasets. There are various domain specific NER systems in existence, such as in medical fields and in different languages. The problem with contracts and other legal data is that there are not much publicly available annotated datasets. Some researchers, such as Ilias Chalkidis et al. [12] proposed NER on contract dataset. They created a benchmark dataset of 3500 contracts by manually annotating them with 11 types of labels. In addition, they also used 75,000 unlabeled contracts for the purpose of word embedding. They introduced various extraction methods to extract labels from the dataset. But because of the security issue, they provided the dataset only in encoded format<sup>5</sup>. They have implemented various methods from Machine Learning methods such as SVM and Logistic Regression [12] to the Deep Learning methods such as LSTM, Bi-LSTM with Conditional Random Field (CRF) [13].

Another recent project by Hendrycks et al. [14] has developed an annotated dataset named Contract Understanding Atticus Dataset (CUAD) on various legal contracts with the help of law students. The main advantage of CUAD is that it is publicly available. The project developers annotated 41 unique labels from the dataset. This is a great step for supporting researchers in this area. They used the question answering task of NLP for identifying key entities from the contract. This is achieved by using transformer models. For implementing this model, they modelled the dataset as Question Answering task such as SQuAD 2.0. i.e., for each label in the contract, the substring which specifies that label is highlighted. This dataset and model is tried in our proposed method.

Most researches related to contract management automation till now are in the area of legal contracts mainly to help the contract analysis job easier for the lawyers. Our research on contract management automation in supply chain domain for smart contract creation is first ever of its kind. This will help the supply chain industry with the blockchain technology, to act smarter and faster.

### 3 Supply Ledger Use Case

Within the Supply Ledger project<sup>6</sup> on the design and implementation of a blockchain platform for a railways manufacturing Supply Chain we are currently developing the automation of contract entity extraction using transformers and transferring of the

---

<sup>3</sup><http://nlp.stanford.edu/software/CRF-NER.shtml>

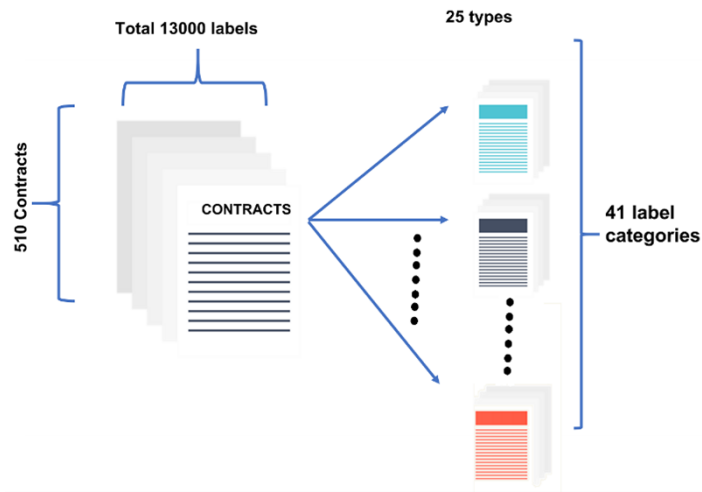
<sup>4</sup><http://spacy.io>

<sup>5</sup><http://nlp.cs.aueb.gr/>

<sup>6</sup><https://www.supplyledger.qa/>



automated contracts to Smart Contracts. In an initial work we have used a Petri-Net-based formalism to model the smart contract workflow in the supply-chain context [15]. We focus currently on the specific part of contracting workflow, based on the CUAD dataset described in [14]. This dataset contains various legal agreements from the Electronic Data Gathering, Analysis, and Retrieval System (EDGAR) by the US Securities and Exchange Commission (SEC). Though for the proposed work, we need only the Supply Chain related contracts; we are considering the whole set as initial experiment and for methods training purpose. The corpus contains 13000 annotations over 510 contracts, with 41 categories or unique labels in this dataset. Fig.3. picturises the dataset. Table 1 shows some of the categories and descriptions given by CUAD.

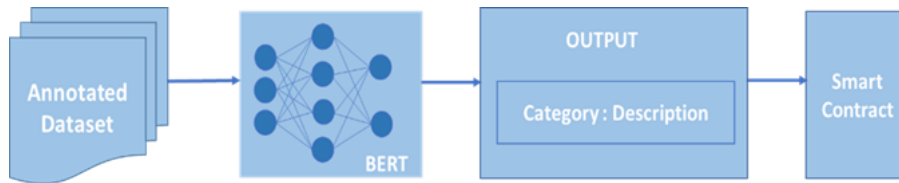


**Fig. 3.** CUAD Dataset description

**Table 1.** Some of the categories and their description given in CUAD

Category	Description
Document Name	Name of the contract
Parties	Two or more parties who signed the contract
Agreement Date	Date of the contract
Effective Date	On what date is the contract effective?
Expiration Date	On what date will the contract's initial term expire?
Renewal Term	What is the renewal term after the initial term expires?
Notice to Terminate Renewal	What is the notice period required to terminate renewal?
Anti-Assignment	Is consent or notice required if the contract is assigned to a third party?
Governing Law	Which state/country's law governs the interpretation of the contract?

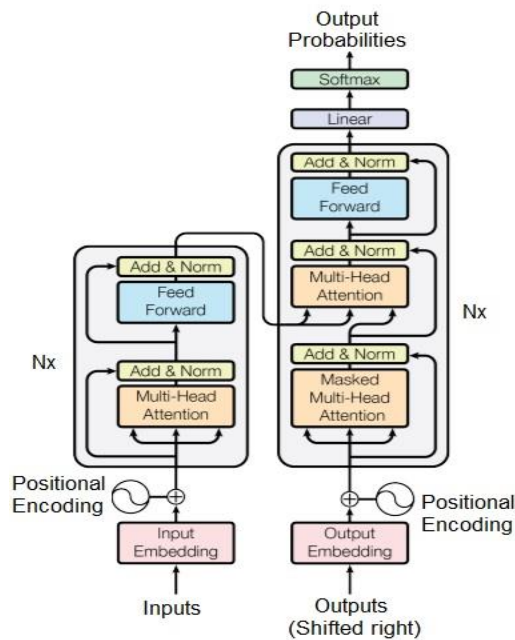
Question Answering model of Hugging Face Transformers library [16] has been used for implementing the model as in CUAD [14]. For each category it treats the label as a question and the answer is the category name with short description. Clauses or entities are extracted using BERT, experimented using python as a programming platform. The overall structure of the model is shown in Fig.4.



**Fig. 4.** Proposed Architecture

### 3.1 More On BERT

BERT is a transformer based deep-learning technique for NLP tasks[17][20]. It is designed to pre-train unlabelled text for bidirectional representation of the data by conditioning both left and right contexts. The pre-trained BERT model has been fine-tuned for downstream functions, to be used in a wide range of applications. The BERT model architecture is shown in Fig. 5.



**Fig. 5.** BERT Model [20]

It is a multi-layer bidirectional encoder architecture. The main attraction of BERT is the self-attention layer. Attention mechanism decides which word has more contextual contribution to the current word. BERT has shown promising results on eleven NLP tasks such as NER and question answering and NER.

There are mainly two steps involved in BERT:

- Pre-training: The model is pre-trained over large number of unlabelled data for different NLP tasks. The pre-training includes two unsupervised tasks.
  - Masked Language Models: In masked representation of pre-training, BERT masks 15% of input tokens at random and predicts the masked tokens from the unmasked tokens.
  - Next Sentence Prediction: This task is aimed to learn relationships between the sentences.
- Fine tuning: The model is initialized with pre-trained parameters and fine-tuned for specific tasks over these parameters. The self-attention mechanism encodes bidirectional cross attention between two sentences. For each task, a specific input-output is plugged in to BERT and then it is fine-tuned with the corresponding parameters. For token level tasks, such as question answering and NER, the tokens are fed into the output layer. For every sequence in BERT, the first token is a special classification token called [CLS]. For classification tasks, the [CLS] token is fed into the output layer. The pre-training and fine tuning is shown in Fig.6.

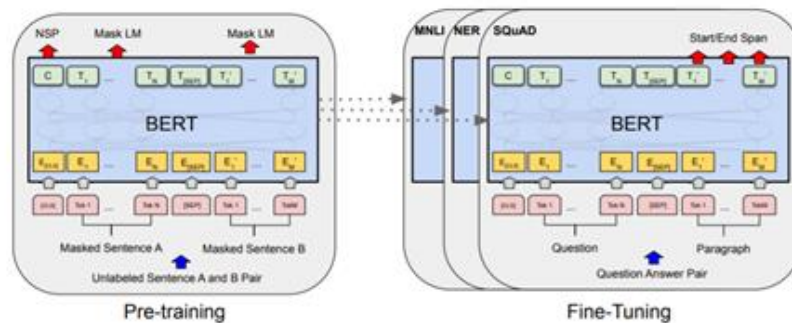


Fig. 6. BERT Pre-training and Finetuning<sup>7</sup>

## 4 Result and Discussion

At the current stage, the design and implementation of the initial steps of global platform is finalized. Contract annotations based on NLP techniques is a tedious but necessary work for the accuracy of our results. The CUAD dataset that is being used now, contains contracts from all legal domains. As an initial experiment, the entire dataset is

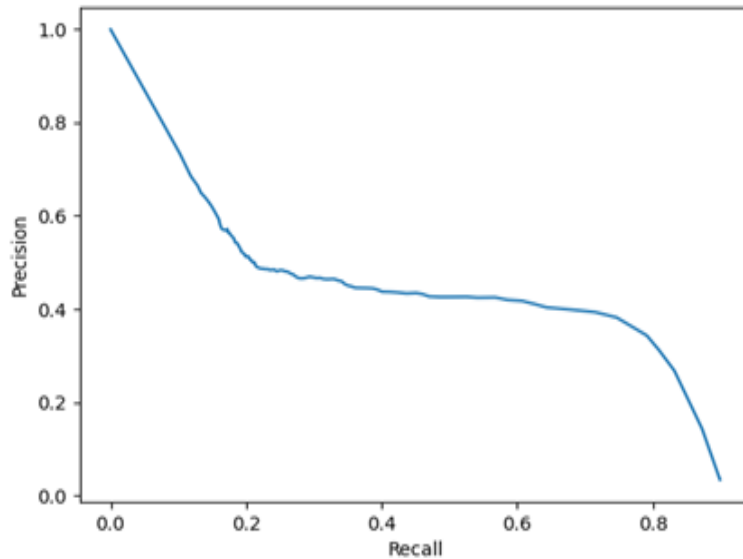
<sup>7</sup> <https://arxiv.org/abs/2009.04968>

used as a whole. While proceeding forward, a minor filtering process has to be performed on the dataset for the selection of only supply chain related contracts. The next step is the finalization of pre-training and fine tuning of BERT on the comprehensive annotated dataset and test it on the contracts provided by the Supply Ledger project's stakeholders.

An initial implementation of BERT for the CUAD dataset has been tried. Fig.7 shows the precision-recall curve obtained for BERT implementation using CUAD dataset. This experiment is carried out as a Question Answering task based on the CUAD dataset structure.

The research is in its initial phase and further experiment with NER on the contract dataset is planned to compare the performance of Question Answering Task and NER for opting best method.

The main attraction of Transformer models such as BERT is that the processing of inputs is not sequential. As the name indicates, BERT reads the inputs bidirectionally, i.e., the reading of entire sequence is performed at once. This feature allows them to parallelize and scale the processing compared to other models. Thus, Transformers can learn the context of the text deeper, and hence, they perform much better compared to other deep learning techniques. This is the reason behind the selection of BERT model for this research.



**Fig. 7.** BERT Pre-training and Finetuning

Next phase of the research is the implementation of Smart Contracts based on the best results obtained from the entity identification and extraction. This will allow the SCM to recommend best Smart Contract template for supply chain related natural language contracts and thus make the entire job easier, cheap, and efficient.

## Conclusion

Smart Contracts are an essential part for blockchain based Supply Chains. These are computer programs that are executed automatically when preconditions are verified. This paper has discussed the characteristics and lifecycle of Smart Contracts and their importance for modern Supply Chain management. The conversion of normal contracts in Supply Chain to Smart Contracts makes the execution of blockchain based Supply Chains more efficient and safer. For contract automation process, entity extraction technique is designed and implemented from annotated contracts dataset using Bidirectional Encoder Representations from Transformers (BERT) approach. The CUAD dataset of 510 different contracts is used in this research and the clauses are identified using the Question Answering task of NLP. We are planning further implementation using NER task on the contract dataset and compare the result to opt the better method. Our next focus is on conversion of the contracts to Smart Contracts based on the extracted clauses. The automated selection and recommendation of best Smart Contract template corresponding to a particular supply chain contract will make the block chain process much easier and cost effective.

## Acknowledgement

This research is part of the National Priority Research Program (NPRP) research project: NPRP11S-1227-170135, funded by the Qatar National Research Fund (QNRF).

## References

1. Koulu, Riikka. blockchains and online dispute resolution: Smart Contracts as an alternative to enforcement. *SCRIPTed* 13 (2016).
2. Fekih, Rim & Lahami, Mariam. Application of Blockchain Technology in Healthcare: A Comprehensive Study. 10.1007/978-3-030-51517-1\_23 (2020).
3. Jabbar, S., Lloyd, H., Hammoudeh, M. Blockchain-enabled supply chain: analysis, challenges, and future directions. *Multimedia Systems* <https://doi.org/10.1007/s00530-020-00687-0> (2020).
4. S. Huckle, R. Bhattacharya, M. White, N. Beloff. Internet of things, blockchain and shared economy applications, *Proc. Comput. Sci.* 98, pp. 461-466 (2016).
5. Belhi A., Bouras A., Patel M.K., Aouni B. Blockchains: A Conceptual Assessment from a Product Lifecycle Implementation Perspective. In: Nyffenegger F., Ríos J., Rivest L., Bouras A. (eds) *Product Lifecycle Management Enabling Smart X*. PLM 2020. IFIP Advances in Information and Communication Technology, vol 594. Springer, Cham. [https://doi.org/10.1007/978-3-030-62807-9\\_46](https://doi.org/10.1007/978-3-030-62807-9_46), pp 576-589 (2020).
6. Christidis, K., Devetsikiotis, M. Blockchains and Smart Contracts for the internet of things. *IEEE Access*, 2292–2303 (2016)
7. Assam Hammi, Abdelaziz Bouras. Towards safe-bim curricula based on the integration of cybersecurity and blockchains features. , Valencia Spain. (2018)
8. C. Sillaber and B. Waltl. Life Cycle of Smart Contracts in Blockchain Ecosystems, *Datenschutz und Datensicherheit - DuD*, vol. 41, no. 8, pp. 497–500, doi: 10.1007/s11623-017-0819-7.( 2017)

9. Nick Szabo. The idea of Smart Contracts. Nick Szabo's Papers and Concise Tutorials, (1997).
10. Dannen, C. Introducing Ethereum and Solidity; Springer: Berlin, Germany (2017).
11. D. M. Lambert and M. C. Cooper. Issues in Supply Chain Management, Industrial Marketing Management, vol. 29, no. 1, pp. 65-83, Jan. (2000).
12. Chalkidis I, Androutsopoulos I, Michos,: A Extracting contract elements. In: Proceed-ings of the 16th International Conference on Artificial Intelligence and Law, London, UK, pp 19–28 (2017).
13. Chalkidis I, Androutsopoulos I. A Deep Learning approach to contract element extraction. In: Proceedings of the 30th International Conference on Legal Knowledge and Information Systems,Luxembourg, pp 155–164 (2017).
14. Hendrycks, Dan & Burns, Collin & Chen, Anya & Ball, Spencer. CUAD: An Expert-Annot-ated NLP Dataset for Legal Contract Review (2021).
15. Rahman, Mohammad & Khalil, Ibrahim & Bouras, Abdelaziz. A Framework for Modelling Blockchain based Supply Chain Management System to ensure soundness of Smart Contract Workflow. 10.24251/HICSS.2021.675. (2021).
16. Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In Proceed-ings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics (2020).
17. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova.: BERT: Pre-training of deep bidirectional transformers for language understanding, NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186 (2019).
18. Badji, Inés.: Legal entity extraction with NER Systems. Thesis (Master thesis), E.T.S. de Ingenieros Informáticos (UPM) [http://oa.upm.es/view/institution/ETSI= 5FIinformatica](http://oa.upm.es/view/institution/ETSI=5FIinformatica) (2018).
19. P. Sun, X. Yang, X. Zhao and Z. Wang.: An Overview of Named Entity Recognition, 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, pp. 273-278, doi: 10.1109/IALP.2018.8629225 (2018).
20. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I.: Attention is All you Need. *ArXiv, abs/1706.03762*. (2017).