

## TOWARDS SOFTWARE FOR CREATING SMART CONTRACTS BASED ON BUSINESS RULES USING NLP

Shynkarenko D.V., Kopp A.M., Orlovskiy D.L.

*National Technical University «Kharkiv Polytechnic Institute», Kharkiv*

In modern information technology, various intelligent technologies have become the most relevant area. Systems based on artificial intelligence have found wide application in a large number of systems designed to perform a wide range of tasks. One of the types of such intelligent systems is intelligent information systems (IIS) designed to support human activity and information retrieval. However, the work of IIS faced the problem of processing large amounts of unstructured data that were not suitable for computer understanding. An appropriate solution to this problem was the transformation of data using smart contracts.

The problem of the development of a software tool for smart contract generation using natural language processing (NLP) tools becomes relevant because, with bigger globalization of the world, more and more issues appear due to the inability to formulate business rules concisely and clearly and then recording them to be accessible by all parties involved. This software should have a convenient user interface to be the intermediate layer between the system and the end-user. Users should be able to create an account in the system and then use the credentials to authorize it. The input of the business rules in natural language should be done using components on the user interface. The system should be able to save the business rule to the database for history and generate smart contract code based on the input using natural language processing. Then the user should have the ability to use the interface for deploying generated smart contracts to one of the supported blockchain networks. Users should have the ability to interact with deployed smart contracts. The use case diagram (see Fig. 1) presents the current scope of use cases for the proposed software. For this work, we propose to use Ethereum-based networks and Ethereum Virtual Machine-compatible smart contracts [1].

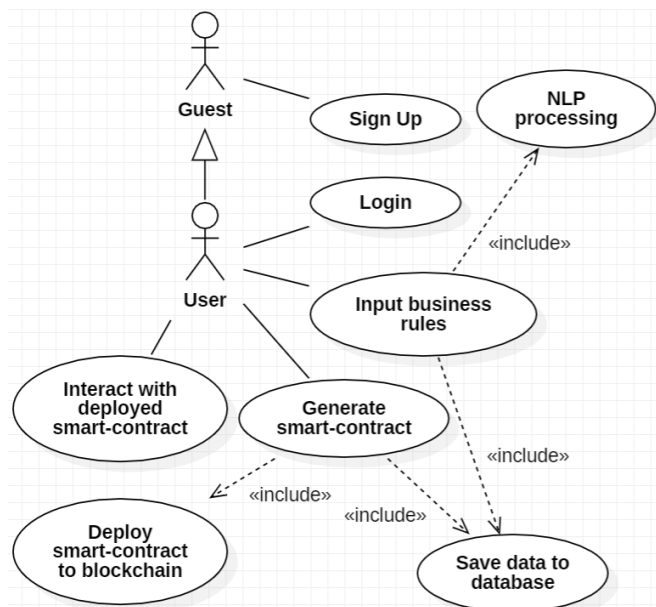


Fig. 1. – The use case diagram of a software tool

### References:

1. Hirai Y. Defining the ethereum virtual machine for interactive theorem provers. *International Conference on Financial Cryptography and Data Security*. Springer, Cham, 2017. P. 520-535.