BLOCKCHAIN BASED “PROOF OF EXECUTION”

Iosif Peterfi[1]

[1] *Ethernity HODL UG, Benediktbeuern, Germany*

The computational science field is growing rapidly multidisciplinary. It already spans across numerous disciplines such as archaeology, biology, chemistry, material sciences, economics, engineering, finance, forensics, history, informatics, intelligence, law, linguistics, mathematics, mechanics, physics, sociology, statistics.

As the scientific process integrates various computational tasks, most of the time they run inside third party environments, using shared resources. Technically the datacenter operators have full access to examine, and tamper with potentially sensitive data during the entire computational process. Mechanisms have been put into place to prevent such situations, however they require very high operational overhead. The auditing of these mechanisms is complex and few people possess the skills required to effectively carry out such auditing. .

The main challenge in this situation is how to easily identify with accuracy and certify the first computational tasks that led to a discovery beyond state of art. To solve this challenge someone must use technical means to ensure high data integrity and confidentiality during the process of the computational task’s execution. Moreover, ensuring programmability of the process similar to the Infrastructure as Code model is desired to be easily deployable with minimal operational costs.

The “Proof of eXecution” method covers all the technical requirements to solve this challenge. It established the grounds for validating with ease and a high degree of accuracy the execution of computational tasks part of a scientific experiment. The main set of metadata that can be validated comprises of the following: the owner of the task, the timestamp when the task execution was requested, hashes of the executable software applying a specific scientific method for data processing, hashes of the dataset(s), the processor of the task, the timestamp when the task execution ended and the result was generated, hashes of the results. Additional metadata can be added to server purposes such as data and process cataloging or hardware resource usage statistics.

The solution proposes usage of two main technologies which handle the core requirements for executing computation tasks in a transparent and privacy aware manner.

The first technology is the blockchain, which has various applications in data integrity, certification, and validations. Bloxberg (https://bloxberg.org), a consortium of more than 50 international research organizations, created a public decentralized infrastructure to foster integrity in research. Validator nodes ran only by vetted research organizations provide a public blockchain with a high degree of trust for all the research and academic organizations around the world. One of the tools provided by the consortium is a certification API that allows a researcher to certify the existence of a piece of data at one point in time. Afterwards, validation of the certification can be performed with ease by anyone at any time.

Widespread technologies that handle research data management right now include repositories which enforce data integrity through hashing. While this works well with repositories that are mutually trusted, sometimes research collaboration is hindered by finding a common trustworthy repository. Using Bloxberg, the blockchain can be used as a central place for storing the hashes to aid the integrity checks during data processing. This covers one important aspect of the requirements of the solution to solve the challenge – the high integrity of the data.

The second technology proposed by the solution is using a trusted execution environment (TEE) enclave. This environment is provided by chip manufacturers as an isolated area for executing binary code inside modern chipsets. The important features of this environment are that it prevents the hardware operators from reading/tampering with the memory space while the tasks are executed, even if they have physical access to the hardware. The environment requires the binaries to be provided unencrypted but supports introduction of secrets inside the execution environment as parameters. The aim of the solution is to use openly certified binaries to be ran as tasks inside the TEE. Confidential datasets would be passed as secrets to the trusted execution environment. Because of the way the TEE woks, if the binary is modified, the secrets will not be decrypted, therefore this ensures no one can modify the process or read/modify the datasets during execution. This covers the second requirement - data confidentiality.

Using various diagrams and explanations, the presentation shows the workflows to maintain the integrity and the confidentiality of the data during the whole process. There are two actors involved in the process: the researcher, also referenced as the data owner, and the datacenter operator, also referenced as the data processor. The workflow of a task execution is handled by blockchain transactions using a smart contract.

First the data owner submits a blockchain transaction describing the task that will be executed which includes all the required metadata. The data itself is uploaded to a shared location. The hashes of the data are saved inside the transaction metadata.

Part of the shared data is the task itself which ideally is a programmable way of generating a binary fileset, such as a docker container or similar. This covers the third requirement of the method to provide a viable solution - tasks programmability. The container would be publicly available so anyone, including the data processor, can examine the method of processing the data. The confidential dataset is encrypted and uploaded.

The open binary fileset includes several methods to integrate the TEE execution with the blockchain. One method is checking whether the execution occurs within the TEE. Another method parses a secret file which includes a location and credentials on how to reach the full dataset, and where to upload the results. Another method checks the hashes of the downloaded files against the blockchain information. A method that handles results is executed inside the TEE. The results are hashed, and the metadata is sent within a blockchain transaction. Lastly, a core method defined inside the open binary fileset is the upload of the files to the location specified inside the secret file.

The data processor sends a blockchain transaction advertising its availability of resources. A match between the data owner request and the data processor request occurs through a blockchain transaction. Then the task is approved for execution by the counterparty, as well as through a blockchain transaction.

The data processor then executes the task as defined in the request. Going through the methods defined in the open binary fileset, a result transaction is generated. After the execution is finished, the TEE enclave is automatically destroyed.

The data owner can now prove how and when the execution took place by having anyone examine the blockchain. This examination can be performed by dApp which is outside the scope of the presentation. This method greatly discourages plagiarism and fraud as somebody can easily prove they ran a specific task at a specific point in time.

When it comes to ethics and transparency in academia and research is important to maintain a clear audit trail in various processes which then can easily aid in proving the data processing took place, even publicly where it makes sense, without exposing the actual data. Then the owners of the data and the processes, as well as auditors can very easily verify and validate such processes. This method can have multiple various applications ranging from managing confidential data such as student grades to complex scenarios which involve compiling statistics that require a high degree of transparency for the process that generated them.