PROOF: Decentralized Platform for Verifiable Outsourced Computation

***Abstract*—Decentralization is reshaping our digital landscape, promising improved control, security, and accessibility. This demo paper explores the potential of decentralization in Cloud Computing, presenting PROOF, a platform designed to enable task outsourcing to any - potentially untrusted - computational resource without compromising the credibility of the produced output. Implemented as an Ethereum dApp, PROOF handles task allocation, execution, and verification, leveraging Blockchain technology for transparent and secure transactions. It utilizes an auction mechanism to match outsourcing requests to resource providers, the IPFS peer-to-peer file system to serve as storage, and Docker containers to facilitate secure and standardized task execution across a network of individual providers. During the demonstration, the attendees will have the chance to interact with PROOF (a) as clients, requesting resources, delegating task processing to remote infrastructures and verifying the validity of results and (b) as resource providers engaging in auctions, performing computations either in an honest or in a malicious way and automatically receiving payments.**

I. Introduction

In the modern digital era, Cloud Computing has become a cornerstone of Information Systems infrastructure, offering scalable and on-demand computing resources. Traditionally, industry giants have dominated this space, acting as trusted entities for the transfer, storage and processing of user or company data. This oligopoly has often raised concerns about the transparency in pricing and resource allocation as well as the sovereignty over data and processing on top of them.

Amidst a global shift towards decentralization, mostly observed in finance and supply chain sectors, the competitive landscape of Cloud Computing is starting to undergo a transformative evolution towards becoming a more democratic and self-regulated market, where anyone can act as a provider, offering unused CPU capacity, or as consumer, leasing the provided resources and outsourcing computation. In such a setup, the challenge lies in identifying a secure method to ensure the correctness of the outsourced tasks by employing publicly verifiable proofs. Existing approaches either target specific applications [1] or have limited applicability due to the restrictions they pose on the type of computation supported [2] or the overhead of the adopted processing framework [3].

To that end, we present *PROOF*, a *P*latform for *R*eliable *O*utsourced *O*perations and *F*unctions, which offers a Blockchain based approach for decentralizing Cloud Computing services, enabling users to delegate computational tasks written in Java to a decentralized network of providers. Through an auction mechanism, providers of computational resources submit bids for task execution, and clients select providers based on a combination of the bid offered and their performance history. The system ensures on-chain the integrity of the off-chain task execution and provides a secure payment process upon successful completion.

PROOF has been implemented as an Ethereum dApp and is publicly available as open-source[1]. This demo showcases the use of PROOF through an easy-to-use web UI, which will allow attendees to act either as clients, submitting a task for computation over a remote infrastructure and validating its output, or as providers, executing tasks. In any role, users can choose to exhibit malicious behavior, attesting the ability of PROOF to detect it.

II. System Design

A. Overview of PROOF

In general, PROOF accommodates two types of actors, namely *Clients* and *Providers*. A Client is an entity in need of computational task execution. Clients provide the task in the form of a compiled Java class, initiating an auction process to identify a suitable Provider. Upon receiving bids, the Client selects the desired Provider. After the computation execution is completed, the Client processes payment and obtains the results of the calculations. A Provider is an entity possessing computing resources and aiming to perform computations for Clients. Providers participate in the auction by bidding for computational tasks. Once selected by a Client, Providers execute the computation within a secure environment.

In the realm of decentralized Cloud Computing, PROOF addresses two pivotal challenges: ensuring the honest execution of a Client's work and safeguarding the Provider's environment. Traditional methods often involve intricate and costly verification processes, potentially discouraging users. Moreover, the execution of unknown code on a provider's machine poses significant security risks. To tackle these challenges, PROOF employs two mechanisms:

1. Verification mechanism: Instead of relying on elaborate and resource-intensive computational proofs, PROOF adopts a streamlined approach using a predefined Java class. Clients submit their code as a Java Jar, embedding a compiled Java class named *Code*. This class must incorporate two essential public methods: (a) *getComputation*, encapsulating the logic for task computation and (b) *getVerification*, returning a string specified by the Client, which is unknown to the provider. The verification string's digest is provided by the Client as a hash and stored on the blockchain before the task execution

[1] https://github.com/Traigor/Decentralized-Cloud-Computing-A-Blockchain-Approach

(upon the auction initiation phase). This serves as proof of the actual execution of the computational task, ensuring that providers cannot directly access or alter the Client's code, thereby enhancing security and transparency.

2. Execution of the computation tasks in a Docker Container: To ensure the security of the Provider's machine and maintain the integrity of the execution process, computational tasks are executed within a Docker container. This containerized environment isolates the execution from the Provider's primary system, preventing any potential harm from malicious code. In addition, it ensures a consistent execution environment and prevents any external interference, verifying that the computational task is executed as intended.

B. System Architecture

The architecture of PROOF's dApp is meticulously designed to facilitate seamless interaction between Clients and Providers, ensuring the integrity and security of computing operations. Figure 1 depicts the architecture of PROOF as a diagram of interconnected components.

At the core of PROOF's design lie two Ethereum smart contracts, each providing a distinct functionality: (a) the *AuctionsManager*, which initiates and manages auctions, handling deadlines and bid-related transactions, and (b) the *TasksManager*, which manages computational task life cycles, ensuring execution, verification, and payment procedures. More precisely, the *TasksManager* implements and executes the verification mechanism described above. In this process, the verification string submitted by the Provider through the *getVerification* method is hashed and compared to the Client-declared hash stored in *AuctionsManager*, thereby verifying correct task execution.

Furthermore, the smart contracts handle the depositing of monetary collateral by Clients and Providers alike, to ensure task commitment. Upon the successful completion and verification of a task, smart contracts facilitate the transfer of compensation from the Client to the Provider, based on the agreed Gwei per second rate. This integral role of smart contracts ensures transparency and security in financial transactions, creating a system that guarantees fair compensation for Providers. Simultaneously, it acts as a powerful incentive, promoting dedication from both parties.

In the Provider's backend, PROOF uses Docker containers as its execution environment, with Java for task execution and Node for transmitting results to the Ethereum blockchain. As a storage substrate, PROOF relies on IPFS [4] for the secure sharing of sensitive data between Clients and Providers, enhancing decentralization.

The frontend for both the Client and the Provider consists of a Graphical User Interface (GUI), which enables auction initiation, bid submission, and task management. It interacts with the Ethereum network through smart contract events.

III. Demonstration Description

The demonstration will allow attendees to interact with PROOF both as Clients, offering them the opportunity to initiate an auction for the outsourced computation of a task, verify the correctness of results and complete payments seamlessly, and as Providers, making bids and executing tasks.

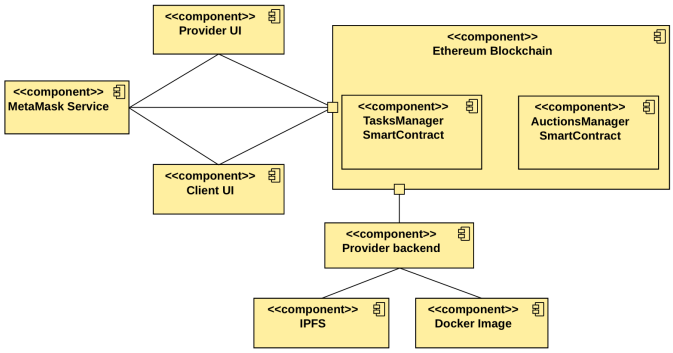

Fig. 1: Component Diagram of PROOF's dApp

Clients begin by defining and compiling a Java class with *getVerification* and *getComputation* methods and then upload it to IPFS, obtaining the CID. Using the PROOF dApp interface, clients initiate auctions, providing essential details such as the IPFS CID of the Java class, the expected verification string, the auction deadline and the task execution deadline.

Through the same web application, Providers bid in gwei per second for task computation. Clients select a provider through the dApp interface, submitting the required collateral. The chosen provider activates the task, committing the necessary collateral and orchestrating the execution process through their integrated dApp. This involves retrieving data from IPFS, running a Docker container, and executing the task.

Upon completion, the verification string and the result are sent to the *TasksManager* smart contract for validation. Correct verification leads to performance upvotes and the return of monetary collateral. The results are uploaded to IPFS, Clients are notified of the total cost through the web application, and payment is automatically made through the smart contract. Once the payment is completed, Clients can retrieve the CID of the computational results and access them on IPFS. In case of malicious Provider behavior, where verification fails, the execution deadline is exceeded or results have not been sent on time, the Provider is downvoted, resulting in collateral loss.

All these operations can be performed through the user-friendly PROOF web application, which provides an intuitive interface for both the Client and the Provider and visualizes their interactions.

References

[1] "Golem," https://golem.network/.
[2] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *arXiv preprint arXiv:1506.03471*, 2015.
[3] T. Bakogiannis, I. Mytilinis, K. Doka, and G. Goumas, "Leveraging blockchain technology to break the cloud computing market monopoly," *Computers*, vol. 9, no. 1, p. 9, 2020.
[4] J. Benet, "Ipfs - content addressed, versioned, p2p file system," 2014, original IPFS white paper.