TeachMePlease Blockchain: Implementation Details

TeachMePlease

November 15, 2017

Abstract

This document describes the implementation details of the blockchain proposed in the TeachMePlease whitepaper. We present an overview of the main entities, their roles and incentives to support the network. Please mind that the project is a work-in-progress and the descriptions provided are subject to change.

1 Architecture overview

Due to the nature of the platform, it has to operate on sensitive data, such as courses, assignments, solutions and scores. Permissionless blockchains, like Ethereum or EOS, would require disclosing this data to the public, whereas the permissive ones, like Hyperledger, lack public verifiability. Our architecture splits the blockchain into two layers: the private layer contains sensitive data, and the public one contains the information necessary to validate the integrity and authenticity of the private blocks. The key entities of the proposed blockchain architecture are presented in Figure 1.

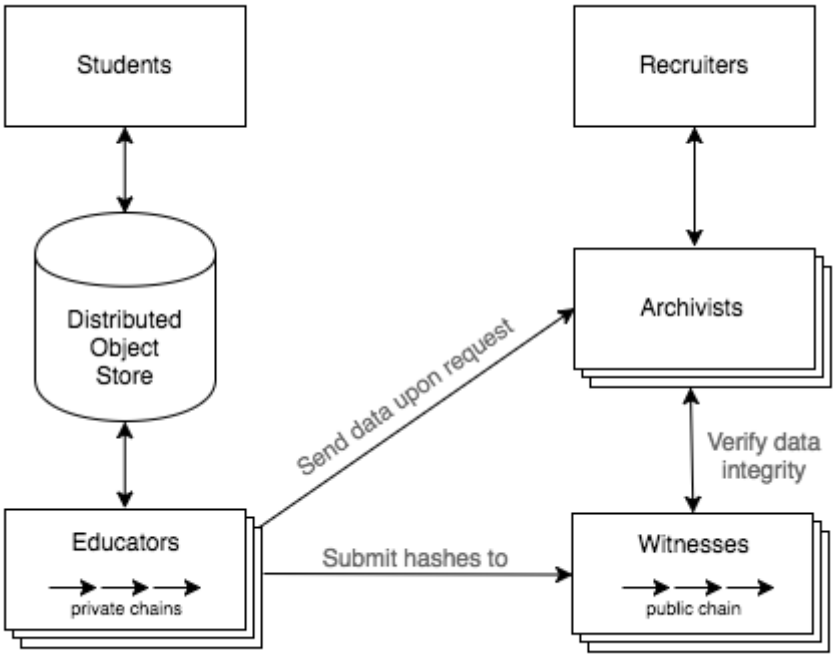

Figure 1: Key entities of the TeachMePlease blockchain

The private layer is maintained by each Educator independently of others. Educators can be either large educational institutes, capable of running their own nodes, or the TeachMePlease platform that runs the chain for the self-employed teachers and small institutions. This layer contains the personalized information on the interactions between the students and the Educator. All the interactions, such as getting an assignment, submitting a solution or being scored, are treated as transactions in the private chain.

Students get access to the platform through web and mobile applications. Using the applications they choose Educators, enroll in courses, get assignments and submit solutions. The scores and the criteria of whether the Student has finished the course successfully are determined by the Educator. The education process from the platform's perspective is as follows:

1. A Student chooses an Educator and a course that she wants to enroll in
2. If the course is offered on a pre-paid basis, the Student uses her app to pay the fee.
3. The Student's application communicates with the Educators' software and performs the key exchange.
4. During the course, the Educator provides assignments that the Student has to complete in order to get the score. The Educator's software encrypts the assignments with the key the parties agreed upon, and puts it into a distributed Object Store.
5. The Student acquires the assignment, completes it and puts the solution to the Object Store (encrypted as well).
6. The Educator then scores the solution and transacts the artifacts along with the score to the blockchain.
7. Upon the completion of the course, the Student acquires a final score based on the scores she got for her assignments. This final score is also added to the Educator's chain.

Making the Educators' chains private opens the possibility for Educators to tamper with the data in their chains. To overcome this issue and make the private transactions publicly verifiable, we introduce the second, public, layer of the blockchain. The public part of the network consists of Witnesses – the special entities that check the validity of the blocks produced by the Educators. They do so by writing the private block headers into the public chain. The Witnesses agree on which public blocks are valid using the specified consensus rules.

Archivists are entities that provide an interface for the Recruiters to communicate with the platform. The Archivists act like a bridge between the Recruiters and the Educators: they choose the institutes that are relevant for the particular request, orchestrate the data disclosure and provide the evidence on the validity of that data. They obtain this evidence by communicating with Witnesses and comparing the headers of the private data blocks with the headers stored in the public Witnesses' chain.

2 Implementation choices

In this section we describe the proposed architecture in more detail. We present the excerpt on the internal structure of both public and private chains and the reasoning behind these choices. We will elaborate more on these issues in the next versions of this documents.

In order to deduce the internal structure of our system, we will first analyze its use-cases. The overview of the education process is given in section 1. The communication between the Student and the Educator is saved as transactions in the private chain. However, the implementation details of this chain mostly depend on the data disclosure process.

We will start from analyzing this process and determining the main issues that arise from the need to disclose and verify the validity of the private blocks. Then we will propose the structure of the private blocks that regards these issues.

2.1 Data disclosure

The Recruiters get access to the private data via Archivists. The process is as follows:

1. Archivists keep records on all the Educators in the system along with the courses they teach.
2. The Recruiter sends data disclosure request to one of the Archivists.
3. The Archivist selects the relevant Educators and forwards the request.
4. The Educators determines the sizes of the data they are going to disclose.

5. Based on the size of the data, the Archivist informs the Recruiter about the cost of the request.
6. The Recruiter pays the fees.
7. The Educators send the data to the Archivist.
8. The Archivist checks the blocks received from the Educator for the validity using the data from the Witnesses' chain, and forward this data to the Recruiter.
9. The fee is distributed between the affected parties.

To mitigate the risk of secondary market arousal, one should ensure that the most part of the data remains in the private blocks. Thus we reduce the size of the Educators' response by incentivizing the Recruiters to make as accurate requests as possible. We propose the following formula to determine the cost of the request:

$$\text{Cost}(request) \sim \exp(N_{actions}(response)) \tag{1}$$

The value received from the Recruiter is then distributed between the Archivists, the Educators that disclosed the data, and the affected students.

2.2 Activity Type Graph

When a Recruiter makes a request to one of the Archivists, the Archivist has to somehow choose the relevant Educators. Moreover, when an Educator discloses the data, it has to provide as minimal set of entries as possible. This set has to be verifiable, which means that the Educator provides the proof of the data validity along with the data being disclosed.

In order to achieve these goals, we divide the data that the Educators store to the atomic Activity Types. Each Educator maintains a journal of transactions per each Activity Type that the Educator offers.

All the Activity Types are grouped into courses that are further grouped into larger entities such as subjects and areas of knowledge. This grouping can be stored as the Activity Type Graph G_A with the following properties:

1° G_A is a directed graph:

$$G_A : \ \langle V : \{\text{Vert}\},\ e_{out} : \text{Vert} \to \{\text{Vert}\} \mid \text{rest}\rangle \tag{2}$$

2° Each vertex of G_A is associated with depth:

$$G_A : \ \langle d : \text{Vert} \to \text{Int} \mid \text{rest}\rangle \tag{3}$$

3° Law of pointing down:

$$G_A : \ \langle v \in e_{out}(u) \implies d(v) > d(u)\rangle \tag{4}$$

4° G_A has special *etc.* vertices:

$$\forall\ v \in V\ \exists u : (u \in e_{out}(v) \wedge e_{out}(u) = \emptyset) \tag{5}$$

The example of the Activity Type Graph (ATG) is shown in Figure 2. The vertex v of the graph is a *leaf* if $e_{out}(v) = \emptyset$. Otherwise we call it an *internal vertex*. Every internal vertex of the graph has a special *etc.* child (some of these are ommitted in the figure).

The need for *etc.* vertices arises from the fact that not all of the Educators teach courses exactly in leaves — some of them offer general courses that provide just the necessary background. For example, some of the universities teaches the basic "Computer science" course, that contains the basics of this discipline. In this case, when the particular category is hard to define, the university would use the $\text{etc}_{\text{ComputerScience}}$ vertex.

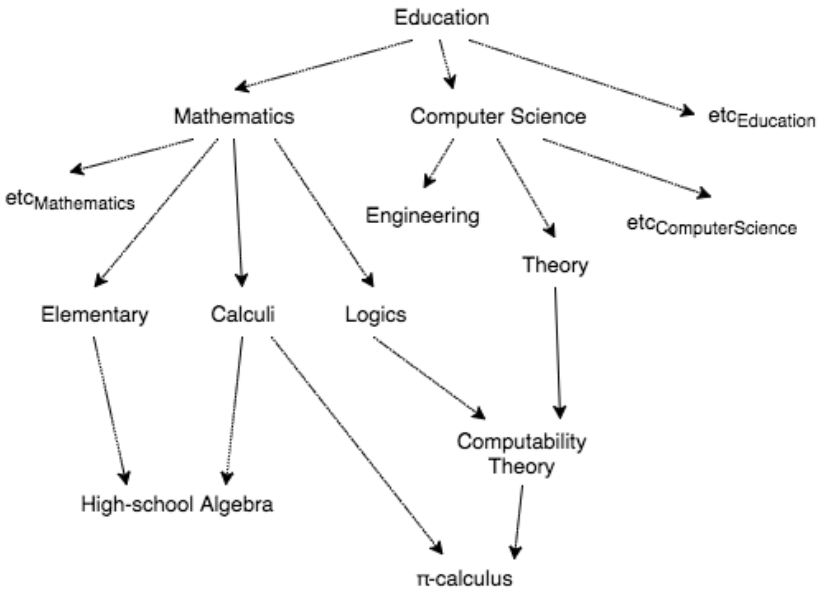

Figure 2: An example of the Activity Type Graph. Some of the vertices are not shown

2.3 Private chain

The structure of the private block is shown in Figure 3. The block consists of a public *header* that the Educators relay to the Witnesses, and the private *body* that remains in the educational institute until it receives a data disclosure request.

During the educational process the Educators emit atomic *private transactions*. These tansactions represent the modifications to the journal of academic achievements in a certain activity type (thus, making a transaction means modifying the journal of a particular course). The transactions can be of the following types:

- student enrolls in a course;
- student gets an assignment;
- student submits an assignment;
- student gets a score for an assignment;
- student gets a final score for the course.

Let us denote an i-th transaction belonging to a certain activity type a_j as $T^i_{a_j}$. The Educators group the transactions that occured during the current block time slot according to the activity type, and construct Merkle trees [Mer89] for these journal modifications:

$$M_{a_j} = \text{MerkleTree}(T^i_{a_j}) \tag{6}$$

The Educator's private block body contains a dictionary of key-value pairs, where each key is the leaf in the activity type graph a_j, and the values are the Merkle roots of all the transactions that occurred in this leaf in the current block. Thus, the private block body is a mapping:

$$a_j \to \text{root}(M_{a_j}) \tag{7}$$

In order to make this mapping easily verifiable, we use a structure called the *authenticated AVL+ tree* introduced in [RMCI16]. This structure is the state-of-the-art research that enables for faster verification of the mapping and allows us to never disclose it: the Witnesses would not have to store the whole blockchain like Bitcoin or Ethereum nodes do. Rather, they would just have to check the private block headers in order to confirm that none of the private blocks were tampered with.

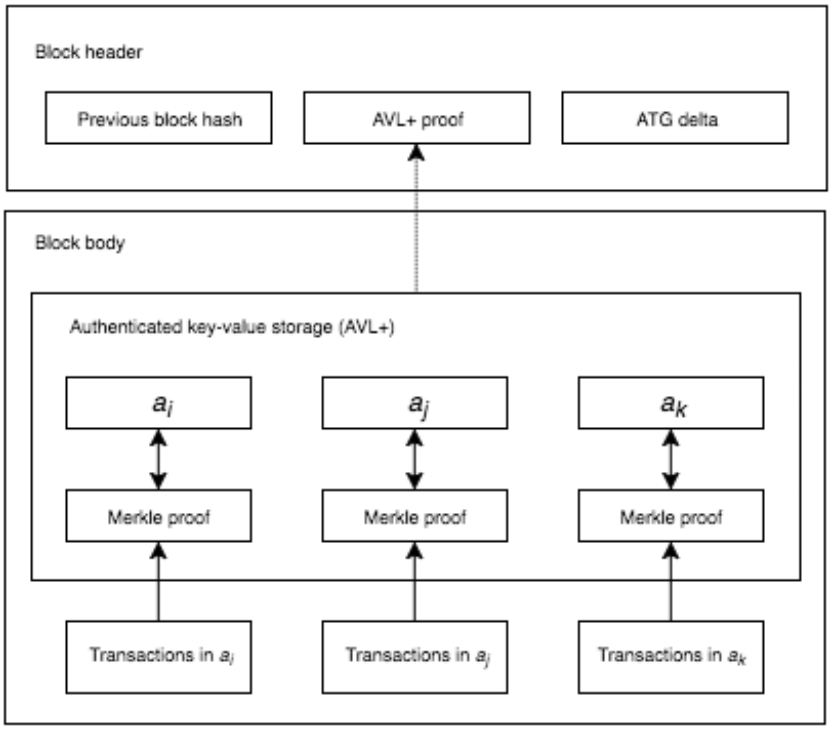

Figure 3: Private block structure

The private block header consists of the AVL+ proof along with the previous block hash and the information on the Activity Type Graph modifications (ATG delta). The *ATG delta* part allows the Educators to inform the Witnesses and the Archivists of the modifications to the courses they teach.

After the end of the block time slot, an Educator signs the block header and submits it to the Witnesses so that the private transactions can be confirmed by the public chain. Thus, the private blocks form a publicly verifiable chain of events, grouped according to the activity type.

2.4 Public chain

The public chain is maintained by Witnesses. They receive the signed block headers from the Educators and make transactions that form a public ledger. This ledger stores the following information:

1. Activity Type Graph and its modification history.
2. Private transaction proofs.
3. Account balances and the history of the value transfer.

The recent achievements in the field of consensus protocols, like the provably secure Ouroboros [KRDO17], allow us to build a public chain based on the Proof of Stake consensus rules. Thus, we can increase the transaction speed and drop the need for the expensive mining. However, with mining being dropped, we need to provide incentives for the Witnesses to maintain the chain and participate in the network.

In order to incentivize the Witnesses as well as the Archivists and the Object Store maintainers we propose a monetary policy with two main sources of income. The first one is the technical pool — a special pool of tokens that are reserved until the participants acquire them through contributions to the operation of the platform. The tokens from the technical pool will be distributed with exponential slowdown. Running the nodes for different entities of the system require different hardware resources and there may be the point where the system lacks the nodes of a certain entity.

To overcome this issue, the complexity and the amount of tokens received by the participants will be determined dynamically so that the equilibrium between the entities is preserved, for example, if the system lacks the Archivists, the incentive to run the Archivit's node would be more then the one for the Witnesses.

The second source of the participants' income is the fees for the transactions in the system. The Recruiters' fee is distributed among all the participants except the Object Store maintainers. The latter obtain tokens from the Students and the Educators paying them for the storage they offer.

References

[KRDO17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[Mer89] Ralph C Merkle. A certified digital signature. In *Conference on the Theory and Application of Cryptology*, pages 218–238. Springer, 1989.

[RMCI16] Leonid Reyzin, Dmitry Meshkov, Alexander Chepurnoy, and Sasha Ivanov. Improving authenticated dynamic dictionaries, with applications to cryptocurrencies. *IACR Cryptology ePrint Archive*, 2016:994, 2016.