Indian Institute of Technology, Kanpur
Undergraduate Project Report

Distributed Academic Grading System Using Smart Contracts

Author: Shikhar Mahajan (150669)
Supervisor: Prof. Arnab Bhattarcharya

13 November 2017

1 Introduction

Okay, so I would like to start from a pointer which is very dynamic in recent times and that is Bitcoin! Bitcoin has taken over the financial sector and has seen an unbelievable rise in its value over the past couple of years. Talking in terms of rupees, it has soared high from merely few paisas in 2008 to about 4 lakhs per bitcoin presently.

So why is it? The prominent feature of blockchain development is commonly understood as the decentralization of financial systems. The main reason behind this is the mechanism of incentivising the block miners for preserving the trust and robustness in public blockchains at the cost of their computation power and storage. Private blockchains, on the other hand are not open to public mining estranging the direct monetary influence. This opens up the scope of this technology to all sectors where data mutability is an existing issue.

(SLIDE CHANGE)

To get straight to the concept of Blockchain Technology, at the core, blockchains offer just another mechanism for the synchronization of distributed databases, there I said it, but is not limited to it. So lets look at the evolution of databases for that matter of fact:

To maintain a database, two entries should not overlap with each other. If the database is distributed rather than centralized, it demands to a higher degree of compatibility leading to Multiversion Concurrency Control (MVCC) where each transaction basically interprets a steady snapshot of the data at a particular time. Hence in MVCC systems, there isn't any foundation of modifying an entry of the database. Rather, it essentially requires back-to-back extended operations of deleting and creating the entry. This is similar to the working of a distributed version control system of our daily use, Git.

Blockchains come in place offering a technology similar to a distributed MVCC in a multi-master replication system. This is implemented by allowing transactions in public space where each transaction block is chained in a time-stamped manner recording each modification to the database.

(SLIDE CHANGE)

Lets us look at the basic terminology related to this technology. A block is defined as a collection of hashed transactions floated in the network at a particular instant of time. The blocks are chained by

storing the hash of the previously linked block address in the freshly created one. The identity hooked to these transactions follows the concept of asymmetric keys. The public address of the receiver of a transaction is embedded inside the output script which only allows the corresponding private key bearer to spend the asset transferred by the transaction.

(SLIDE CHANGE)

Following is the list of industries which are heavily disrupted by the arrival of Blockchain Technology. To throw some facts, it is believed that Blockchains has the potential for reducing bank infrastructure costs to about 30% and the amount the global blockchain market is expected to be worth is $20 billion in 2024. Now this leads to a big question!

(SLIDE CHANGE)

Are blockchains really the solution to every existing problem? Let us dive deeper into this. Let us look into the type of problems that should be considered solving by Blockchains, or rather some common characteristics associated with all of them.

(SLIDE CHANGE)

2 Blockchain- The Solution!?

- Distributed database: The project must involve a shared and distributed database. An entry corresponding to the database refers to the data/asset recorded in the name of the entity owning that entry. Any modifications to these entries are only entertained through transactions which are published globally (in the case of public blockchains) for acceptance by mining nodes.
- Multiple writers: As transactions can be performed by any permissioned node (everyone in the case of public blockchains), it involves multiple entities transacting modifications to the database.
- Mistrust among the writers: The concept of consensus is relevant only when there are multiple writers present with conflicting goals/records. This leads to the structure of forks in the blockchain, resulting into global acceptance of the longest chain in the fork.

SLIDE CHANGE

- Interactive transactions: Transactions essentially involves deleting and creating entries in a collective bilateral fashion. This accounts to modifying existing entries in the database linking each transaction that involves a particular entry.
- Disintermediation: This is a fundamental fracture point of many projects aiming to exploit blockchains. It is often the case that the project doesnt actually need the removal of a trusted central party. Disintermediation leads to delays in acceptance of transactions and involves the cost of applied computation power and storage of all the verifiers involved.

SLIDE CHANGE

- Permissioned miners: The provision of permissions in blockchains opens up its scope to many fields. This is a primary pointer to the development of private blockchains where an authority in control of the blockchain permits some particular nodes to mine blocks. This margins the protocols adopted by private banks with a public protocol like Bitcoin.

SLIDE CHANGE

One important note here is the fundamentally understand the completely "new" features that are introduced with the blockchain technology. People say that the concept of smart contracts ie running a piece of code to update the blockchain state but I am afraid to say that centralized databases already offers stored procedures that does the same thing. People say that blockchain has the unique append only ledger but again centralized databases can be customized to do this. It is just like allowing only inserts rather than deletes in the language of SQL.

So what exactly new is offered by this technology!? The answer is "disintermediation". I had pointed out this before as the main fracture point of many of the projects looking to adopt Blockchains.

There is one more difference that goes out of favour of Blockchains and that is Confidentiality. It should also be noted that in public blockchains, although a direct personal identity is not linked with the public address, an adversary can track transactions involving a particular public address as all of them are broadcasted to the network and are accessible by every node. So, this leads to a trade-off of decentralization achieved at the price of confidentiality. This is also one key takeaways of the presentation I believe.

SLIDE CHANGE

So it essentially boils down to these 3 genuine use cases of blockchains namely lighweight financial systems, provenance tracking and multi party aggregation. The first point is pretty much self explanatory considering the whole lot of cryptocurrencies and credit systems being developed today. Provenance tracking is essentially the tracking down the ownership of the asset associated with the blockchain. Examples of exploitation in this sector can be seen in healthcare industries and real estate to trace the origins of medicines or lands or households etc. The third sector is multiparty aggregation whose prime motive is to just keep the records intact. This is mostly a permissioned blockchain and confidentiality is not an issue here.

SLIDE CHANGE

So talking about all these stuff about the non-fiscal aspects of Blockchain technology, lets move closer to our demonstration of distributed academic grading systems. But before that just a quick introduction to smart contracts might help!

SLIDE CHANGE

3 Smart Contracts

The main reason of evolution of smart contracts was that people started embedding metadata in bitcoin transactions to serve purposes like digital assets and document notarization. Now, this led to a development of many public blockchains for this specific purpose. And then came Ethereum, which offered a single blockchain platform to built all distributed applications on top of it. Two types of accounts are linked to Ethereum, one user accounts and the other smart contracts. I mean like you send money to account in a financial system, you can now send some data say an input to the piece of code linked to that smart contract account.

SLIDE CHANGE

Here are the main constituents or say features of a smart contract:

- It expresses the business logic as computer programs.

- It represents events which trigger that logic as messages to the programs, this is the same as input that I cited before.
- It uses digital signatures to prove who sent the messages ie it makes use of assymetric keys for identification.
- It puts the programs, messages and signatures on a blockchain, recording all of them.
- It executes every program for every message on every node.

Out of these five constituents the first four are well and good but not the fifth one. This is actually the only bad side of smart contracts. It involves a lot of re computation leading the wastage of power. To avoid this, people are working to develop a concept called as proof of stake rather than the proof of work.

SLIDE CHANGE

So now that we know the full background, lets move on one of our implementations of smart contracts. Let us first quickly list the conditions that should be met while implementing an academic grading system.

SLIDE CHANGE

- A student cannot fake his/her identity to get enrolled in the course twice.
- A grade can only be assigned by the professor and once assigned, cannot be changed by him/her at any later point of time. In case of any discrepancies, the chairperson is the only authority who can alter the grade.

  SLIDE CHANGE

- Each student has the permission to view his/her own grade but not someone elses.
- Everyone in the network can do statistics on the course grading scheme such as percentage of students getting an A grade, failing the course etc.
- Professor is able to justify the grading scheme if questioned later(Off-chain data). This essentially involves storing hashed values of pdfs or scanned images onto the blockchain network.

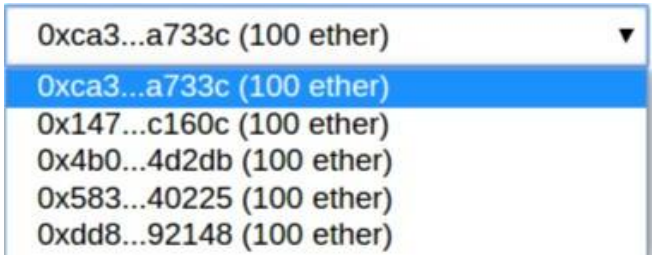
  SLIDE CHANGE

4 DAGS

The smart contract used for this demonstration is written in Solidity language and is deployed in Remix, a web browser based Integrated Development Environment that provides basic User Interface and Javascript virtual machines to test our contract.

4.1 Deployment

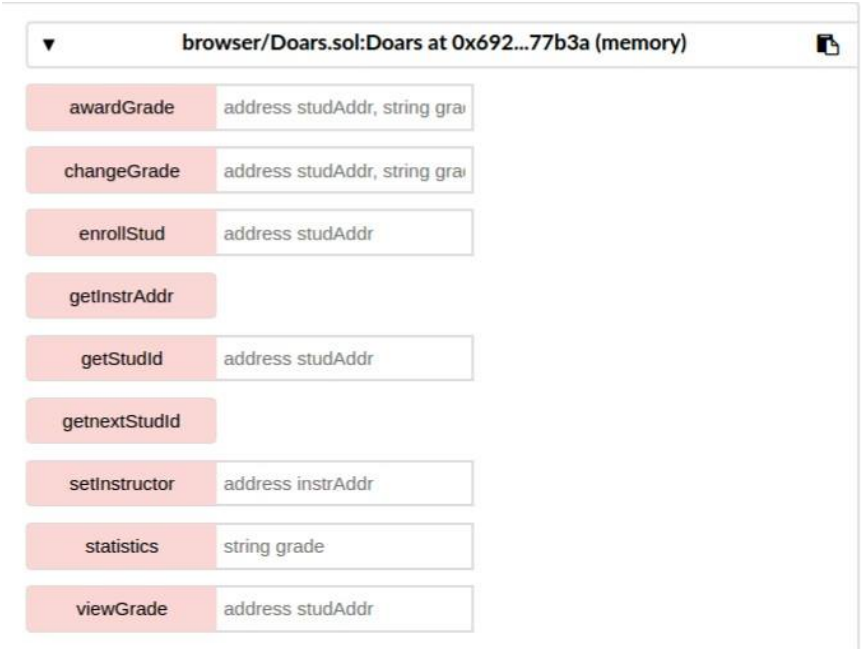
Remix gives us 5 accounts with 100 ether each that can be accessed by the smart contract for testing purposes. We distribute these 5 addresses into chairperson, instructor and 3 students getting enrolled in the course.



The chairperson creates the smart contract by specifying the course ID and the maximum number allowable students in the course. The chairperson can also bind the contract to a special account address if needed. By default if this field is left empty, it is set arbitrarily to any available contact account address.

4.2 Functions

Following is the image of the UI provided by the Remix IDE to perform the contract's functions:

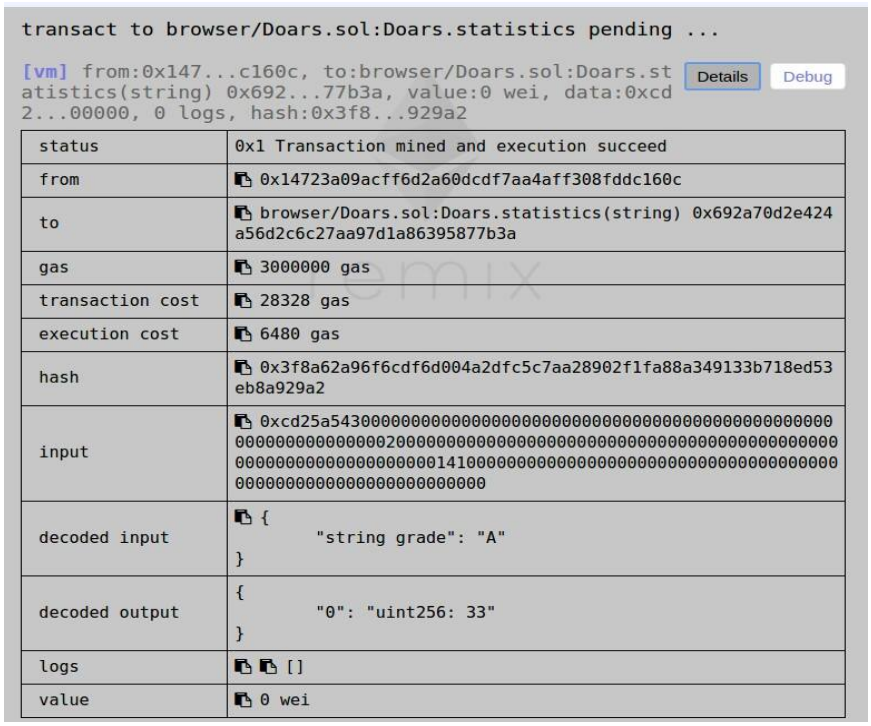


The description of each function is as follows:

1. **setInstructor:**
   (a) Takes as input the public address corresponding to the instructor
   (b) Requires the transaction sender to be the chairperson
2. **getInstrAddr:** Outputs the course's instructor's public address
3. **enrollStud:**
   (a) Takes as input the public address corresponding to a student
   (b) Each student enrolled in the course is assigned an unique number ID starting from 0 in the order of enrolling students
   (c) Requires the transaction sender to be the instructor
4. **getStudId:**
   (a) Takes as input the public address corresponding to a student
   (b) Outputs the student's ID
5. **getnextStudId:**
   (a) Outputs the number which will be assigned as the ID of next enrolling student of the course
   (b) Also equals the total number of students enrolled in the course so far
6. **awardGrade:**
   (a) Takes as input the public address corresponding to a student and the grade string assigned to him/her
   (b) Requires the transaction sender to be the instructor
7. **viewGrade:**
   (a) Takes as input the public address corresponding to a student
   (b) Requires the transaction sender to be the instructor or the student himself/herself
8. **changeGrade:**
   (a) Takes as input the public address corresponding to a student and the changed grade string assigned to him/her
   (b) Requires the transaction sender to be the chairperson
9. **statistics:**
   (a) Takes as input a grade string
   (b) Outputs the percentage of students in the course getting that grade

4.3 Transactions

We will look at how a transaction is interpreted by the smart contract upon execution of a function. Take the function *statistics* for instance. Here is an image showing the detailed description the transaction created by this function when given the input string as "A":



transact to browser/Doars.sol:Doars.statistics pending ...

[vm] from:0x147...c160c, to:browser/Doars.sol:Doars.statistics(string) 0x692...77b3a, value:0 wei, data:0xcd2...00000, 0 logs, hash:0x3f8...929a2 Details Debug

status	0x1 Transaction mined and execution succeed
from	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
to	browser/Doars.sol:Doars.statistics(string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
gas	3000000 gas
transaction cost	28328 gas
execution cost	6480 gas
hash	0x3f8a62a96f6cdf6d004a2dfc5c7aa28902f1fa88a349133b718ed53eb8a929a2
input	0xcd25a543000000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000000000000000000000000000141000000000000000000000000000000000000000000000000000000000000000
decoded input	{ "string grade": "A" }
decoded output	{ "0": "uint256: 33" }
logs	[]
value	0 wei

- *status* shows the current status of the transaction in the blockchain
- *from* shows the transaction sender's public address
- *to* shows the contract's account address in the blockchain
- *gas* shows the upper gas limit spent on the transaction specified by the sender
- *transaction cost* shows the gas utilized in mining the block associated with the transaction and appending it to the blockchain
- *execution cost* shows the gas utilized by the contract in computing the result and returning it as the output
- *input* shows the hashed input script
- *decoded input* shows the argument name, type and its value given to function, "A" in this case
- *decoded output* shows the return result index, type and value of the function, 33 in this case.

4.4 Future Work

- Scale the current distributed model to accommodate many courses instead of one in a single blockchain.
- Modifying the current model such that a student gets an unique ID among all students which is consistent in all courses that he/she gets enrolled in.
- Introduce off-chain data to include hashes of the soft-copies of all the answer scripts.
- Migrate the application completely to the Truffle framework from browser based Remix IDE to make the system adaptability easier.