



The blockEDU Protocol

A Peer-to-Peer Education Network

August Xth, 2017

Nathan Ginnever

Dylan Lott

Abstract

blockEDU is a protocol built on-top of the Ethereum blockchain and a network of distributed reputation where data is stored in a private and decentralized cloud. blockEDU implements a consensus mechanism that combines lessons learned from bitcoin, backfeed [cite backfeed], and UniTEC [cite uniTec] called Proof of Affordable Education (PoAe) that rewards academic contributions. It is an alternative education protocol that can host free--and open source--course management systems (CMS) where Distributed Open Collaborative Courses (DOCC), Small Private Online Courses (SPOC), Massive Open Online Courses (MOOC), and institution free traditional universities can be explored. The goal of blockEDU is to solve some of the apparent issues with current accredited universities as well as online courses in a way that makes quality and reputable education affordable and accessible to all by determining value in decentralized protocol. blockEDU offers the ability for anyone to create custom courses or structure degree programs that conform to current academic paths without the centralized trust records and identity provided by the current academic institutions. We leverage the underlying blockchain consensus on transactions to maintain the state of the blockEDU virtual chain that incentivize nodes to maintain the community's consensus on the value of education.

1 Introduction

The question has been raised as to whether institutional universities and online course systems are continuing to provide an affordable and quality education for students. The increase of student loan debt to income ratio has caused many students to begin looking for alternatives. Alternative forms of education enabled by the web have become a popular industry in the past decade as students face increasing tuition rates and the inability to afford traditional accredited universities. With MOOCs you do not need to be geographically close to your institution of learning. As long as the academic community has an internet connection and a cursory knowledge of technology they may connect with each other to advance knowledge. Online institutions such as Udacity, edX, and Coursera have seen success as a perhaps viable ways of spreading mass education at an affordable price. They promise to offer affordable and accessible information to those who otherwise would not have had a chance to learn. blockEDU is an attempt to aid in Goal 4 of the 2030 Agenda for Sustainable Development where the UN has envisioned a way to ensure inclusive and quality education for all and promote lifelong learning. It is encouraged that all existing education platforms, online and offline work together to achieve this goal. One can imagine traditional universities such as Harvard and MIT and online systems such as the MOOCs listed previously committing to this consensus and building their governance model on-top of blockEDU.

We present a new consensus model called *proof of education* (PoE) that has blockchain characteristics that incentivise educators to provide affordable education. This is accomplished via miners on the blockEDU protocol curating in progress courses, earning EDU tokens as an incentive to determine the value of a lesson. A system of smart contracts is used to verify that students are eligible to enroll in courses as well as provide the state machine for ongoing courses. We provide pointers to a distributed network of storage that insures private and permanent course records. Course content is always uploaded to a public cloud that is financed by a federation of organizations interested in keeping information permanently accessible.

2 Problem Statement

Education has become a booming industry fueled by capitalism and privatization. We have come to call this *the industry of education*. What was once thought to be a service to society has become a system where companies can take profit from a core need. Students and teachers rely on centralized institutions to provide an understanding of trust. Who should a student trust to provide a quality or highest tier education or who should a teacher trust to be a worthy student in their course? As institutions both online and offline take on higher numbers of students, there is a lack of personal interaction between educator and student and students to other students that is important to building relationships that facilitate knowledge. Not

everyone learns at the same pace or in the same way, therefore it is a challenge to provide a template to teach mass numbers of students at the same time.

Germany has made their public universities essentially free, which is a great step in the right direction, but this says nothing of the quality of education they're offering. As we intend to show, the blockEDU model can not only incentivize the creation and dissemination of affordable education, but can also offer a way to gauge the quality of that education, while allowing a system to discourage bad actors.

2.1 The Profit Model - Industry of Education

The industry of education is run just like every other industry, with profit as its goal. Organizations that once started with a non-profit business model have inevitably moved towards institutionalization, advertising, and profit. These evolutions toward a profit are methods to sustain the organization, whether that be attracting better educators and affording better tools, or sustaining the business. There is an argument that the free market and private education incentivises the best educators and provides the best tools to the select few that can afford it. Preserving these qualities outside of capitalism is a goal presented by the UN and we offer a bootstrapping method to get there.

2.1 (a) The Online Education Profit

Every system has a cost to building and maintaining its infrastructure and systems such as the platform provided by edX try to encourage the open source community to do this service to reduce costs, however their open source system is hard to run. The fact that these systems are too complicated to set up for an average educator it enables the creation of an entire industry of cloud service providers to take profit from students and teachers. In order to maintain the software, these online course providers populate their systems with advertisement. Developers and maintainers should be paid fair wages for creating these systems, but online universities and course management systems fall into the same pattern as traditional universities where CEOs and CTOs are taking large salaries at the cost of both the student and educator. There must be a reasonable cost to developing these organizations that is not being explored.

2.1 (a) The University Profit Model

Offline private institutions are also failing to keep courses affordable for students and are facilitating the rise of online education. Current institutions have too many auxiliary fees that don't benefit the student such as administrative fees. Higher up positions such as deans and CEOs are taking multimillion dollar salaries [source open information on actual salaries]. The increase in student loan debt discourages students from enrolling in traditional universities. [source student debt information].

2.3 Accreditation

Universities maintain their place in society because there is nowhere else for teachers to have the reputation they need to be trusted [back this statement up, seems reasonable but may not have evidence].

Until recently it has not been understood how trust can be accomplished without a centralized authority to issue it. [more research into the need for accreditation, question why it has to be granted from a university. Trust models will be discussed further down in the rep system. We will be describing how we can offer trust without a central authority. Perhaps create an analogy to CA problems vs web of trust]

2.4 Lack of Interactions, Feedback and Community in MOOC

MOOC lack a flexibility that allows the community to decide how courses should be structured. Some universities such as Amherst and Duke have taken hard stances against using MOOC and have expressed concern that the massive nature of MOOCs or the need for those involved to understand technology or a lack of a strong sequenced structure makes them unacceptable. They often forget about foreign language speakers and only cater to English as they are not able to coordinate on a local level. These MOOC systems fail to explore the difficulties of relevance and knowledge transfer across borders.

The ability to scale class sizes up is potentially one of the largest weaknesses presented by traditional MOOCs and institutions. There is often a lack of interpersonal connections and diversity in scope that cater to all types of learning. Because of this there are large incompleteness rates and only a select few that will succeed from MOOCs. Others will drop off and lose money to a for-profit institution. Large class sizes also limit the ability of the educator to learn better ways of structuring their courses in the future. Teachers often report that personal feedback is a valuable resource for both educator and student.

2.5 Learning Models

[let's discuss standardized tests, and the lack of value that letter grades give in understanding someone's knowledge]

[cite algebraic reasoning] A case example of the value of personal feedback can be found in teachers and researched beliefs about the development of algebraic reasoning in high school students. In this example the teachers must organize their mathematics testing and problem sets by their predicted difficulty for students. Textbooks sometimes are not a reliable source for this reasoning. "The Symbol Precedence Model of development of algebraic reasoning, in which symbolic problem solving precedes verbal problem solving and arithmetic skills strictly precede algebraic skills, was contrasted with the Verbal Precedence Model of development, which provided a better quantitative fit of students' performance data."

3 Proposed Solution

The blockEDU protocol does not define a single method that should be considered best for educators or students. The protocol builds communities that get to decide what is best. We understand that education benefits when there is greater communication and enthusiasm to be a part of a community that progresses the area of knowledge. Our protocol can not dictate that a course should remain small so that personal

attention can be achieved, but we can offer a system that makes this possible at the choice of every educator and student. We strive to make a global system that knows no boundaries and can express knowledge at a local level relevant to those involved. blockEDU is a protocol, thus it is possible to be used in anyway necessary.

3.1.1 Censorship

The blockEDU chain runs on a public ledger that is redundantly copied and shared among the nodes in the Ethereum network.

3.1.2 Quality of Information

3.1.3 Dissemination and Spread

3.1.4 Data Persistence

3.1 Profit Solution

Decentralization of institutions that provide knowledge will reduce costs incurred for maintaining the institution. This is done by removing the need for maintaining a centralized facility. blockEDU removes the ability for institutions to persue activities that are not directly related to their core mission. The protocol and community governs the institution rather than an operation of individuals with high paid positions. Faculty members are paid by their perceived worth from the community and thus there is no need for a governing body to determine if faculty funds are being used appropriately.

3.1.1 Online Model

blockEDU offers an easy to use platform for educators and students such that there is not a need for unnecessary technical installations or over priced cloud data solutions. This reduces the cost burdened to both educators and students that were once forced to pay for services to run a CMS. This eliminates the need for an industry of service providers that have been working with institutions in the industry of education. While there are still monetary costs to using blockEDU, it is expected that these fees will be less than cost of scaling a course on traditional online education platforms.

3.1.2 Offline Model

There are no C-titles with large salaries here. A public and transparent record is available from the non-profit for each of the salaried employees in the organization. We do not require the maintenance of sophisticated buildings to achieve the same caliber of education provided by private institutions. Anyone can set the meeting points for courses where it may not be necessary to have all of the equipment needed for a chemistry course paid for by students taking a single math course. Removing the physical institution

offers a flexibility never seen in a quality education. We encourage world renowned educators in all disciplines that do not require a laboratory or gymnasium to teach a course in a local library or park.

3.2 Consensus

To encourage that education remain free, we present a consensus model that *rewards educators for maintaining a free course* as well as nodes for *reporting correctly on this consensus*. Just as bitcoin [cite bitcoin] maintains consensus on the correctness of financial transactions, blockEDU maintains consensus on the idea that education should be affordable. The consensus protocol incentivises educators to not charge students for their service. While giving education and course content away for free is not mandatory, the protocol rewards teachers that maintain this standard, and does not reward those that present a cost for their time and content. [backfeed] “These mechanisms ensure a fair distribution of the generated value to each individual, according to the perceived value of their respective contribution to the organization as a whole.”

3.3 Accreditation

We understand that blockchains don't do everything well and certainly are costly. It does not always make sense to store information on every machine in a network that does not need this data, however there is a compelling argument that blockchains provide publicly verifiable information that cannot be altered when statements need to be made. If used appropriately, this can be a powerful feature of a network of nodes agreeing on such a statement.

One of the benefits of institutions is the ability to stamp approval on a record that others can trust to be accurate. We propose that we can replace this institutional trust in a student's records, or an educator's reputation with a blockchain. Digital signatures within a blockchain offer a way for identities to broadcast their reputation and value without the need to consult centralised institutions since we can trust the statements made by owners of these private keys linked to identities.

3.3 (a) Educator Reputation

The blockEDU protocol creates the ability to use distributed reputation in a way that provides the accreditation needed for students to trust that they are getting a quality education. Educators can use this proof as a way to demonstrate their value to other educators and for side organizations that encourage the development of knowledge in their specialty. We discuss further in section 5 how educators gain their reputation in our system. These records are also stored in a blockchain so there will always be a pointer that anyone can view to determine if they are interacting with a trusted educator.

3.3 (b) Student Reputation

Educators now have the ability to create a permanent log of signed records that students have completed courses that anyone can verify. Students are given a reputation and are given a database to maintain their

course records to pass knowledge forward to future employers or education groups to prove they have achieved a quality education. Each community member has their course records permanently printed in the Ethereum blockchain. Students maintain these records on a distributed cloud storage platform with encryption keys so that they always retain who gets to see their achievements. To prove the accumulation of knowledge to the protocol, smart contracts read course completion registers that allow them to unlock more advanced courses. In this way you could imagine building entire degree paths on-top of the blockEDU protocol.

3.4 Personal Interactions

The possibility of peer-to-peer education will ensure that the community always forms the organization. There will be no centralized authority to dictate how a course should be sized, scheduled, or monetized.

Choice is a core tenant of blockEDU and if other organizations are not comfortable creating a decentralized community, they may choose to establish a more traditional centralized and permissioned university role. You may set the permissions on your course contracts to allow anyone to join or only those that you deem fit or that have completed enough prerequisites. More information on this is presented in section 7.3.

Just as massive course structures are possible so are small localized communities with personal interaction. Educators choose the max number of identities that may register in the course contract. If they wish to keep a course small it is as simple as limiting this number and structuring the course to use the frontend CMS forum application to facilitate personal communications. You may also use the system to advertise local meetup facilities that will happen at predetermined times. Ultimately the members of this community get to determine where they belong and what works best for them.

4 blockEDU Protocol

Here we present how the blockEDU protocol is structured. The protocol is inspired by the blockchain system first laid out by bitcoin [cite bitcoin]. blockEDU can be seen as a related to the work done by backfeed [cite backfeed paper] protocol differing in that it is not general distributed governance but instead specific to education.

4.1 Consensus - Proof of Affordable Education

Analogous to bitcoin's proof of work scheme (PoW), proof of affordable education (PoAe) will be the protocol for educators to receive financial incentives. Students have the ability to give a reputation to each individual course taught so that there is a mechanism for deciding if the course was *of quality* and

affordable. Educators will be rewarded from the total reserves of EDU tokens created much in the same way that mining occurs in other cryptocurrencies. Proof must be supplied in reputation that the course was of value to the students given as a real number in the range $[0,1]$. A monetary reputation is assigned to the course as a separate unit to decide that the course was taught affordably by the students also in the range $[0,1]$. A student may get a good education from a teacher, but still assess a low monetary reputation for the course. If either of those conditions is below their respective thresholds, the teacher will not be paid out from the reserve of tokens.

4.2 EDU Tokens

EDU is a standard Ethereum token and the total supply is held in reserve on the blockchain state. The tokens provide a method for rewarding the educators for maintaining the consensus that education should be affordable. EDU works as a method for students to provide a fee for entering the courses which becomes the only method for educators being rewarded once the total supply has been depleted. Section 4.2.5 provides more information on how the total supply is awarded over time. EDU also enables educators to create courses by incentivising full nodes in the reputation system to report their course reputation to the course contracts.

4.2.1 zEDU Tokens

To maintain an objective that any users in the protocol may remain private, we offer a mechanism for converting traceable Ethereum tokens to zkSNARK Zcash tokens. While still in the early stages of development during the time of writing this, blockEDU will provide a method for atomically swapping the ERC20 token for a future standard Zcash token. Zcash and tokens are deployed as the payment mechanism for the storage network. This is done by [cite sapling, read method]... This enables private payment methods to both students and educators in the event that certain education becomes oppressed or deemed illegal by authorities.

4.2.2 Mining EDU

While there is no computational work or proof of stake for mining to ensure the network reaches consensus, a loose analogy to bitcoins [cite bitcoin] mining can be made in the sense that tokens are still distributed from a total supply when a proof has been established. As stated in 4.1 the proof here is the reputation scores of the course from student feedback passed in by oracles that are further described in 4.4.1.

4.2.3 EDU Blocks

A course block is similar to bitcoin where miners create blocks of transactions to be rewarded from the total supply. However, instead of blocks filled with financial transactions or state updates, educators are creating courses filled with students receiving knowledge. The proof attached to each block, or course, in

this sense is the reputation ratings derived by the students and the ability for the system to check that the course was provided in an affordable and quality way. During the mining process in bitcoin, mining hardware computes hash values until it hits a certain target. In blockEDU, teachers will produce informative education until their reputation score for doing so is recognized. The reward for these blocks also mimics bitcoin by varying over time to maintain a deflationary model in token distribution.

4.2.4 Block Transaction Fees

Transaction fees are an optional parameter set by the student before entering a course. Instructors may choose to pick up students that provide a higher fee in EDU for entering the course. This is analogous to a bitcoin transaction fee, where a miner is rewarded an extra amount for picking up the transaction into their block. When the total reserve of EDU runs out in [calculate time it runs out] this will be the only remaining reward for educators.

4.2.5 Token Distribution

[formulate exact details on the reserves and the deflation of issuance, create a graph plotting the issuance over time]

We control the distribution of tokens to maintain a timeline of token depletion. Transaction fees will be the only mechanism for paying educators when the total supply runs out, and we speculate that this may not be a feasible mechanism to maintain the consensus just as Bitcoin will face uncertainty in its mining protocol in the approximate year 2140 when the block reward stops. The answer is not know but blockEDU may update the protocol in the future or rely on the transaction fee for entering a course as a method for balancing affordable education.

Just as in bitcoin, we devise a method similar to that of difficulty adjustment to keep the total distribution of tokens at a relatively constant rate. The bitcoin protocol attempts to control the time it takes to mine a block by increasing or decreasing the hash target, thus changing the computational power needed to hit the target. In blockEDU, there will be a variable number of courses active at any given time, which will cause the distribution to become variable.

Another factor to inconsistent supply depletion is the average reputation an educator has and how much they will be paid for each class as is described further in 4.2.6. The issuer contract keeps track of the distribution of tokens over time. This is done by the *checkIssuance()* function that can be called by anyone. As in bitcoin, this function call will be available to be called approximately every two weeks. This is done every 2016 blocks in bitcoin, however we witness timestamps provided by the miners to allow this check as block times are quite variable in Ethereum. This call checks the total issued during the last interval and adjusts the max reward per course accordingly.

[supply some distribution stats]

4.2.6 Block Rewards

An issuer contract holds the reserve of all tokens that will be issued to educators. During four stages of the course contract, the issuer will rely on nodes in the reputation network reporting on the course like an oracle to determine if the educator should be rewarded for maintaining consensus. The course contracts has a time-to-live (TTL) for the course length as provided by the educator creating the course via a CMS. Each course contract has four states that will open their contract up for rewarding the educator from the issuer contract. This is built into the constructor of valid course contracts and calculated automatically upon a course being instantiated. This mechanism is to ensure that a teacher may be paid in installments during the length of the course. The educator may send reputation requests through designated nodes handling their course context area to initiate these reports.

The fields that are able to be reported on are an *affordability value* and an *education quality value*. Recommendations are issued by the enrolled students In order for a node to know how to report correctly to a contract during each quarter of the courses TTL. The thresholds for the contracts to pay from the reserve are an affordability score above 0.8 and a quality score above 0.7.

The amount of tokens issued to each educator is already not a static value as mentioned in 4.2.5. Block rewards should also have different values depending on how reputable the educator giving the course is. In this way a teacher with a high value of recommendations is paid more for their time. Teacher context area recommendations come with a value based score between [0,1] as well as textual information about the teacher. The contract can't read the textual information so the value score of the reputation of the teacher will be used to determine how much of the max possible block reward they should get. If Alice has an educator trust rating of 0.8, then she will be rewarded 80% of the highest block reward during each of the quarterly payout rounds. 80% of the fees placed in the course contract by students will also be paid to Alice, and the remaining 20% of the reserve tokens remains in the token reserves and 20% off the fee tokens will be rewarded back to the students. This scheme however will have a bootstrapping problem where the educators global reputation starts with a 0 meaning that all new educators must give their courses away for free. This is encouraged in the consensus protocol so we will maintain this and it also works to prevent sybil attacks.

In order for the contract to get an accurate report on the reputation and affordability scores for the course, it must not take the first report it sees. There is a window of time that each contract can take scores for each payout period. Since not all nodes in the reputation system have the same network graph and pull the same amount of recommendations, there can be an issue with nodes that report from a limited view of what students are recommending. In order to solve this, each report from the nodes comes with not just the average scores from the students, but also the number of recommendations they have seen to compute that average. The contract takes the score of nodes that report with the highest number of seen recommendations.

4.3 Identity

Identity in our system is in the same format as Ethereum as the network is bootstrapped from the RLPx cryptographic network and transport protocol [cite rlp]. This gives our reputation system the ability to sign transactions on the blockchain that correspond one-to-one with the signatures in the reputation system.

Other forms of identity are required to work with the distributed storage network and Zcash tokens (should privacy in token be required). These identities work the same way with slightly different formats, namely the hash functions and base encodings differ.

As to not burden each user of the system and fulfill the requirement that the protocol is as easy to use as current systems, blockEDU provides a service on the network that is entrusted to manage the private keys. In this way, students and teachers can simply remember a username and password, however, it is encouraged that those who are willing to store their own private keys do so. Private keys that are used to issue tokens will not be stored on any nodes. It is the responsibility of the user or the blockEDU organization to input public keys and maintain their token private keys. The blockEDU non-profit org has a centralized service to help facilitate the exchange of tokens to fiat currency to enable ease of use.

4.4 Nodes

Unlike users that solely interact with the CMS on-top of the blockEDU protocol, nodes on our reputation network provide a service to the user community. They *report the reputation records*, they *maintain the reputation database*, they *report as oracles* if they have enough positive recommendations, and *find trust paths* by facilitating user requests so that the ease of use requirement can be fulfilled.

4.4.1 Nodes As Oracles

A specific set of nodes chosen in the blockEDU protocol report to the course contracts on whether or not consensus is maintained. In doing so the protocol rewards the educator for mining the course. This decision is recommended by the context area nodes for that course and determined locally by assessing the recommendations given by the students in the reputation system by the oracle. Each course contract will look at the knowledge contract before allowing a report to update the state of their contract. This is to prevent arbitrary entities from reporting these scores. It is important to incentivise these nodes to maintain the consensus protocol, thus we reward reporting nodes based on their reputation in the system. These rewards are further discussed in section 4.4.3.

As nodes are responsible for storing and maintaining the reputation system in the blockEDU protocol, they make for a distributed oracle that behaves in their best interest of keeping their reputation. If a node does not report the reputation data they are getting correctly to the course contracts, they will receive bad recommendations when receiving feedback for this and be unable to facilitate further reports to the contracts for a time as designed by the UniTEC[cite untec] system. In order to decide what nodes make good oracles, when the CMS looks for a node to facilitate educator requests initially, it looks at the knowledge registry for the *oracles context area* and determines the nodes with the best expertise as an

oracle. A node is randomly selected encouraging nodes to gain high trust to be in this set and therefore receive more requests so that they may pick up more fees.

Oracles are placed in the registry when any request facilitating node reaches the threshold for high enough reputation. Again we have a bootstrapping problem that we solve in this case by providing a few trusted oracles from a set of secured blockEDU organization nodes. These nodes will only decentralize this process in the beginning of the system until further nodes come online with a high reputation. A transparent exit of these nodes is made once there are enough oracles.

4.4.2 Nodes As Request Facilitators

Each node/s is responsible for managing a context area/s in the network. It is encouraged that these nodes are selected by the community for having some interest in promoting the health of their academic area if they manage the academic reputation context areas. They are responsible for connecting each user to the reputation system so it is essential that these nodes be trusted. Just as every identity in the system has a reputation, these nodes have a reputation which ensures that they behave in the best interest of the community. Any user may decide to run a full node and advertise themselves as a facilitator. Doing so gives the nodes the ability to begin building trust in the reputation system. Ideally the entire network would be made up of these full nodes and not end users.

The request facilitator reputation is the recommendations issued to these nodes for providing the service of handling all of the requests passed to them from user identities not running nodes from the CMS bridges.

4.4.3 Oracle Fees

In order to incentives nodes into becoming well behaved oracles in maintaining consensus, the protocol requires a fee from the educator that is initiating the reward function call on the course contract as overseen by the issuer contract. As mentioned in **4.4.1**, nodes with a high enough threshold of reputation in the oracle reporting context area are chosen at random as being eligible for facilitating these requests, thus being eligible to receive oracle fees. The mechanism for achieving this is to contact the knowledge contract to find a node that can act as the oracle, when the block reward is issued, these nodes will receive 10% as an oracle fee. [supply the algorithm for this]

4.4.4 Storage Fees

A token transfer method is created to enable the payment of the distributed cloud storage when requests are made that carry a data upload. This is important due to the cost that these nodes incur for satisfying this role. The nodes must pay for the storage providers in the network to hold the recommendation data and the course material and state data as well as facilitate the bandwidth traffic necessary for the communication between other nodes. It expected that educators and students pay for their storage costs for the duration of time in which the plan to have their data stored. In the case of student course records, where data is meant to be stored permanently, recurring payments will be necessary to compensate the

data miners storing these records. Section 6 goes into further detail of how a distributed cloud intends to be less expensive than traditional cloud storage providers.

4.4.5 Request Fees

Users must also pay miners in the Ethereum and Zcash network for interacting with the smart contracts. It is only required that individuals pay for their own storage. If the node is processing a request for a user, the nodes is incentivised to pay the auxiliary transaction fees as a method for building trust, where being a trusted authority on the network is considered valuable since trusted nodes may receive oracle transaction fees. This is also a sybil attack prevention measure as discussed in section 5.7.

To incentivise nodes to maintain their reputation, an optional *request fee* is given to each request a user makes to a node. These fees are calculated by the CMS on-top of the protocol for each request when a facilitating node is selected automatically so that teachers and students do not have to understand the protocol costs. The teachers and students are presented with these fees before making their requests to the network and can optionally choose to reduce them or not pay them at all. Each node may choose to reject the request and require a fee from the user to cover their costs. Nodes that do not have a high reputation are incentivised to process transactions with no fees to build their reputation. As the network builds trust, the fees remain set to 0 until the majority of the network is comprised of nodes with good reputation.

4.4.5 Offline Nodes

Since a node is the bridge for users to communicate with the network, it is important that there are backups when a node goes offline. Pointers to the location of reputation and course data collected are stored in the blockchain as a way of backing up data loss. Users are prompted to record their private key phrases on paper and keep them secured as a last line of defense in the event that the nodes facilitating their identities disappear. Recovering data will require consulting the data registry contract with the public key of the user's identity. Their private data was encrypted with this key so all data stored in the storage network will still be recoverable.

6 Distributed Cloud Storage

A persistent data storage network is needed for both educators and students to store course content and reputation records.

7 Ethereum Contracts

Ethereum is a decentralized network of nodes that process transactions and maintain consensus on general state. Ethereum is used in the blockEDU protocol to maintain the state of reputation, consensus, and data. It will provide a mechanism for unlocking courses and course content, it maintains the identities of the users in the system, and it provides the registries that provide pointers to the current state of the applications running on the protocol.

7.1 Ethereum Shortcomings

Ethereum is used as the network to provide a decentralized point of authority on the state of the blockEDU protocol. Storing data in the ethereum network is expensive. [gather details on op storage costs per byte of hex data] and every transaction that updates the state of a contract costs a miner fee for computing the new state. There will be a large amount of transactions to this state machine and as a method for offsetting the costs we propose two solutions.

7.1 a State Channels

State channels are a method of offsetting the costs of state updates to contracts. This is done in the same manner that micropayment channels work in the Lighting network of bitcoin [cite lighting]. In the case of state channels, the state is not limited to that of only financial transaction but opened to the broader updates of general state.

The largest amount of state updates to the blockEDU protocol are found when we are updating the knowledge contract for finding RECs, updating the various registries that store the oracles and new courses, and when the data pointers for forums and course related information needs to be updated.

Instead of broadcasting all of these updates directly to the network, we hold on to signed updates locally that are hash locked with their counterparty. This way these signed broadcasts can be sent to the contract when intermediate state can be thrown out. An example of this is when a student and teacher are sending homework assignments back and forth that are stored in the storage network. Instead of broadcasting the pointer to the state head of the data, they will receive the updates on the side network of blockEDU nodes and only broadcast the final state of their homework interaction once the assignment is completed. This also works for the communication forums where students involved in a thread are collectively updating the pointer to the latest state of the thread. Only when a thread is closed will this pointer be sent to the contract as method of permanent storage.

7.1 b Linked Data pointers

The Orc, Swarm, and Filecoin networks provide meta information as to the location of uploaded data in the distributed network. This meta information is updated in the state of the contracts as a way to offset the data storage costs of large data such as written recommendations on the blockchain. These pointers may point to either encrypted personal file locations, or public shared data.

The meta information needed to locate a file on these networks is as follows.

[This is the information needed to find the location of the farmers in the bridge database, I need to research the meta info needed to contract the farmer directly storing the user's contract. Alternatively we can keep these semantics as we are designing bridges. A choice should be made here on how we want to design these pointers]

* @param {String} id - Unique hash ID

[revise this for SSTORE costs]

Each string is a X bytes of X hash algorithm. The maximum registry size for the Ethereum VM is 32 bytes so concatenation of these two fields fits into one register with an MLOAD instruction from the contract state. The total cost for a X byte MLOAD instruction is X units of gas. [fill in the Xs]

```
1  pragma solidity ^0.4.11;
2
3  contract OrcDataPointer {
4
5      address owner;
6      bytes32 pointer;
7
8      modifier onlyOwner() {
9          if (msg.sender != owner) throw;
10         _;
11     }
12
13     /// Create ownership of this pointer to data.
14     function OrcDataPointer() {
15         owner = msg.sender;
16     }
17
18     /// resets the pointer.
19     function removePointer() onlyOwner {
20         pointer = 0x0;
21     }
22
23     /// Update the given pointer to a new state of the data.
24     function updatePointer(bytes32 _pointer) onlyOwner{
25         pointer = _pointer;
26     }
27
28     /// returns the pointer data
29     function getPointer() onlyOwner returns(bytes){
30         bytes memory _pointer;
31         assembly {
32             _pointer := sload(add(_pointer, 24))
33         }
34
35         return _pointer;
36     }
37 }
```

Figure 7.1 Simple contract to store pointers to data networks.

As shown in Figure 7.1 a simple method in a contract can be used to load pointers to data onto and read from the Ethereum blockchain to reduce the cost of storing actual data. The above example uses a permissioned modifier to ensure the community is not able to destroy data pointers, and thus the data itself, if they do not own it.

7.2 Issuer Contract

The issuer contract is the mechanism in which educators are paid for mining and maintaining consensus. It is registered as the owner of the total supply of EDU tokens and has the ability to issue tokens.

The issuer contract contains a mapping of all active course contracts. When an educator initiates an issuance request this contract instantiates the given course contract keyed with the educators identity. It then determines if the reputation system has enabled the educator to receive EDU by checking the course state and reported affordability and quality metric.

Determining the amount of tokens issued to the educator is two-fold. It first reads the current max issuance amount as outlined in section 4.2.5 of the deflationary token model. It then also contacts the knowledge contract to get the overall reputation of the teacher in the range [0,1] as provided by the oracles. The amount awarded is computed as described in section 4.2.6.

[create visual for this simple equation?]

7.3 Course Contracts

The course contracts make up the structuring of a decentralized education. This can be a single siloed course, or these courses may depend on each other to form a more complex sequence.

7.3.1 Course Instantiation

An educator is prompted by a CMS to fill in the necessary fields for creating a course before the contract is deployed to the network by a facilitating node. Here the educator must determine the max enrollment size, the prerequisites required, the Unix epoch of when the enrollment period will end, the Unix epoch of when the course will end, the minimum number of students required, and link to the syllabus stored in the distributed storage network. The course has a data upload that stores the course information beforehand to get the data pointer and autofills this into the request.

When the course is created, a call is made to the global course registry contract that places it in the mapping for advertising it to students. At this point students may begin registering until the designated enrollment period ends. If the course does not meet the minimum number of students as set by the educator, the course contract will self destruct.

[layout course fields]

7.3.1 Enrollment and Fees

When a course is created by an educator, the student may find it via the CMS that consults the course registry contract and attempt to enroll. The student must decide how much of an EDU fee they are willing to pay during this process. There is an enrollment time where the educator is given the ability to choose students who have supplied a higher fee as defined by the provided timestamp during instantiation. There are two mappings in the course contract for enrollment, one for potential students and one for confirmed students.

7.3.3 Content Pointers

Each course contract will have a storage for pointers to the distributed storage network for course content. Educators get a pointer to the course content state for uploading assignments. The students mapping contains addresses to data pointers to the latest state of the student's data. This pointer is updated when students need to upload data to the course such as homework assignments or engage in forum activity. As new assignments are rolled out during the course, the state of this content is updated by the educator updating the pointer in the contract. There are permissions set on these pointers such that only educators and students can update state changes to their respective pointers.

7.3.4 Dependent Course Contracts

Organizations have the ability to create sequential courses for full degree paths. An example being the calculus 1 through 3 series in traditional universities. When creating a course contract, educators have the ability to designate what courses are required by an identity before enrolling. This is done by supplying a list of other course contract addresses to the course in question.

When a student uses their identity to be placed into the enrollment mapping for a course, a preloaded list of course contracts will be consulted. If these students are not stored in the state data for the list of courses as having completed the course prerequisite, their enrollment transaction will fail.

7.4 Registry Contracts

Registry contracts are used as a way for the other contracts in the system to understand prerequisites, the bootstrapping process of the knowledge provider to find recommenders, the total list of active courses, and the pointers that hold each user's distributed file bucket that holds their data.

7.4.1 Knowledge Provider Registry

The knowledge contract will contain registries partitioned by knowledge areas to return experts in the desired area. This is to defeat the bootstrapping problem of a node in the system not being able to find RECs for given context areas. If a node is not well connected to the network, and they have no RECs

stored locally, they may consult this contract as a method of trying to find a node that may have a REC stored locally.

The knowledge contract also stores information on the oracles in the network. It provides the overall registry that each course contract will consult to determine if an identity providing reputation information to the course is valid.

7.4.2 Course Registry

This contract provides the CMS with a way of aggregating all of the deployed contracts on the blockEDU network. Each time the CMS is loaded for the user, this contract will be consulted to provide a view to students of what courses are active to be enrolled in, and what courses have completed and are still open to review course content. If it was decided during instantiation by the educator that the course material should live for free, the course will fall into a completed mapping where students may consult the course contract and find the content data pointer still intact. Assuming the storage network is still being paid to maintain these records, the student will be able to review the content.

This registry is also used by the issuer contract when calculating block rewards.

7.4.3 Public Profile Registry

The public profile contract holds the identity information for the users of the CMS. This data is public names, addresses, profile pictures and textual data. This can be replaced with other standard identity registries like uPort.

7.5 University Contracts (Future Implementation)

While not currently supported by the first CMS application built on-top of the blockEDU protocol, we are pre-planning for a future CMS that is geared toward traditional universities. These contracts are an attempt to build a fully decentralized university that will rival private for profits in their quality of education. Universities are complex organisms so we are challenging the online education model first. It is not impossible for organisations to utilize the blockEDU protocol in it's current state as they may handle their governance off-chain and still bootstrap the reputation system into theirs.

8 Aristotle - The first CMS

The first CMS is built by blockEDU on top of the protocol token, reputation system, and a distributed storage Network. It is not necessarily the only system that can be built. Other organization that are currently in existence are welcome to replace their backend with the blockEDU protocol and auxiliary systems like the swarm distributed storage method presented here. [enter name] serves as an example of this may be done.

A market of courses similar to Udemy is being built here. All courses will be listed with their uploaded information about what the course is. The reputation scores for each educator in the market can be determined by issuing a request in the local view of the course.

The CMS will allow users to log in, check their courses, sign up for new ones, create a course, and more. Forums will foster discussion in specific topics, and links to outside materials will be recognized for faster integration into the student's workflow. For example, GitHub links will auto-pull stats for the repo in question, relevant info about the repo, and other pertinent information. Wikipedia links will pull up a quick preview of the page and quick links to the article's major citations. These features are aimed at making the students study sessions more productive, helping them find information faster and more reliably.

8.1 Client Application Features and Scope

Structure

The client application will be built in JavaScript with the Vue framework, possibly with React in some parts if necessary.

The client application will be a static app, packaged up and bundled using Webpack, minified and kept with performance in mind. Vue is often noted as the fastest framework right now, and offers a good tradeoff between speed of development and app performance.

Deployment

The backend application will be a Node server, serving up a local app of Vue.

The backend application will be deployed in a Docker container, using Rancher to orchestrate container deployments and uptime. This backend will connect to the BlockEDU Bridge [cite next section] to interact with the blockEDU platform.

Data persistence will be achieved with MongoDB at the Bridge layer.

Mobile

We will package up a native implementation for mobile and desktop in either a Vue native framework such as Weex or a React framework React-native.

Authentication

8.2 Requirements

This is a broad, technology-agnostic overview of the requirements of the client-side and server-side applications.

Client-side

- Video upload
- File upload
- Sign up, login, and authentication
- Course management
 - Syllabus creation
 - Milestones in each course
 - Course creation
 - Course deletion
 - Maintenance and editing
- Reputation and feedback management per course
- Course discussion
- Forums and discussion areas
- User reputation
- Administration and moderation

Server-side

- Authentication
 - Key management
 - Passwords
- blockEDU network interface and interaction layer

8.3 Future CMS Implementations

The first CMS will be a marketplace for courses and discussions, which we see as valuable, but short-term. The long-term goal of our platform is to compete with universities and colleges in research, offer degrees provably tied to people, and focus on education as a human right that should be accessible from all corners of the world and to any who desire it.

9 The blockEDU Bridge

Platform Bootstrapping

To fulfill ease of use requirements, the blockEDU organization hosts a bridge server that connects the CMS applications to the blockEDU network. This bridge facilitates HTTP requests from the users not running nodes to the full nodes in the network. As a point of centralization, this is a weakness in the protocol and a temporary solution. Full documentation is provided and anyone can run a bridge in the event that the blockEDU organization is attacked or no longer exists.

Long Term Decentralization

Eventually, we would like to stray away from this centralized bridge model as new methods for handling private keys, browser side encryption, and light client APIs to blockEDU nodes are built, but for bootstrapping the platform, some amount of centralization will have to occur.

We have been exploring the possibility of full nodes running in the browser, this way every teacher and student can participate directly in the network to fully decentralize the reputation network. At the current time of writing this, browsers do not make for good nodes. Mobile applications are not long living nodes in the DHT, there is no stream cipher encryption in browser to allow fast private data uploads, and communication through browsers relies on centralized webrtc servers.

10 Possible Attack Vectors

10.1 Course DDoS

This comes from the ability of educators with low reputation being able to register a large amount of courses and reduce the current block reward for the system. It may be a good idea to require a bond or some course creation cost for educators.

10.2 Educator / Student Collusion

It is possible that teachers could create courses that have little or no meaning and fill these courses with identities that they control. As stated in [] all identities with a low reputation will receive no reward for teaching courses. This will force a malicious actor to build up their reputation slowly to receive block rewards for teaching meaningless classes. There is a cost to the actor for building this reputation in network fees and bonds as an educator. We provide a method for deterring this action by putting a limit on the length of a course. All courses must be active for at least one month. Oracle nodes are also incentivised further to report courses that are of no value to the reputation system for an increased reward of 20% of the block reward.

10.3 Sybil Attacks

Sybil attacks are a common threat to distributed systems where it is cheap and easy to create multiple identities. The most common way to discourage such attacks is to put some cost on the creation of nodes.

The blockEDU protocol has a reputation system to identify malicious participants. However attacks are still possible. An attacker may register a new identity, place themselves in the knowledge contract as a highly knowledgeable identity in many context areas, and once requests are received start issuing false RECs. The disadvantage here to the attacker is the TMC where their recommendations are not yet trusted by the network. An attacker may choose to gain trust by behaving reputably or copying valuable RECs from others and later turn malicious, but this is time consuming. The amount of damage a node could

incur by turning malicious is still costly to the network. The feedback mechanism will detect this switch to malicious behavior, but only after the damage is done.

The reputation system is linked to Ethereum so that there is a payment mechanism to increase the costs of performing bad transactions in the blockEDU network. In order to build trust, the entities must perform actual financial transactions in Ether. There is already a cost incurred by nodes when they have to upload data to the knowledge contract, store pointers, report to the course contracts as oracles or facilitate course contract creation. Therefore building reputation on the blockEDU protocol does cost a real world value that could prevent attackers from gaming the system.

11 Conclusions

[TODO: Get citation sources]

$$\phi\Sigma \subseteq \Omega \sum_{\alpha=1}^{10}$$