The blockEDU Protocol

A Peer-to-Peer Education Network

July Xth, 2017

Nathan Ginnever
Dylan Lott

Abstract

BlockEDU is a protocol token 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 similar to that proposed by backfeed [cite backfeed] called Proof of Affordable Education. 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. BlockEDU offers the ability for anyone to create custom courses or structure degree programs that conform to current academic paths without the centralized trust provided by the current system. BlockEDU will remain an uncontrolled non-profit that has no ability to impose any unnecessary fees on students or educators. There are no C-titles that will take unreasonable profits from this organization, and it will remain a community driven project throughout its lifespan.

1 Introduction

The question has been raised as to whether institutional universities 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 way 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.

2 Problem Statement

[research education outside of the united states, it is clear that the industry only cares for profit here, but other countries should receive credit for doing it better]

Education has become a booming industry [pull stats on education industry]. 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, who should a teacher trust to be a worthy student in their course? As institutions both online 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.

2.1 The Profit Model - Industry of Education

[spend more time expanding this idea of the industry of education, there are a ton of resources here. This needs the most fact checking and is perhaps the most convincing problem statement]

The industry of education is run just like every other industry, with profit as its only 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 non-profit business.

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, populated by advertisement software, and enables the creation of an entire industry of cloud service providers to take even more profit from students and teachers. These systems are too complicated to set up and it is advertised that these providers are creating a valuable service for educators. edX thus creates a proprietary system that has salaried employees to create a quality platform. 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 taking large salaries at the cost of both the student and educator.

2.1 (a) The University Profit Model

Offline 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. Teachers are incentivised to keep course material proprietary as a way to produce income.

[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. 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 Personal Interactions and Feedback in MOOC

MOOC lack a flexibility that allows the community to decide how courses should be structured. Some Universities 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 incompletion 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.

[site 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 Solutions Overview - Why Decentralize?

[overview of why decentralization is a good idea here]

3.1 Profit Solution

BlockEDU offers an easy to use platform for educators and students such that there is not a need for unnecessary technical installations or cloud data solutions. This reduces the costs burdened to both educators and students that were once forced to pay for services to run a CMS. This will eliminate an industry of service providers that have been working institutions in the industry of education.

BlockEDU puts forth an idea that using decentralized technologies can enforce a non-profit model that is self sustaining. The organization is maintained by reserving a portion of the total supply of EDU tokens. Doing so commits the organization to being only worth what the community believes it is worth. As the free market decides to support or not support blockEDU. There will be no C-titles with large salaries. A public and transparent record will be available from the non-profit for each of the salaried employees in the organization.

3.2 Consensus

To encourage that education remain free, we present a consensus model that rewards educators for maintaining a free course. Just as bitcoin [site 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 charge money 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

[write general overview of how distributed systems offer an alternative to centralized trust]

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, and to pass this knowledge forward to future employers or education groups so they may trust that a quality education was provided. Educators can also 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.

3.3 (b) Student Reputation

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. Courses are never limited to the constructs of a few, but rather molded by the masses

Privacy and a lack of censorship are built into the systems used by the blockEDU protocol to avoid potential issues with control mechanisms such as authoritative governments outside of the protocol.

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.

Just as massive course structures are possible so are small localized communities with personal interaction. You can choose the max number of identities that may register in the course contract. If you 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 Protocol

Here we present an in depth look at how the blockEDU protocol is structured. The protocol is inspired by the blockchain system and the work done by backfeed. BlockEDU can be seen as a subset of the backfeed protocol in that it is not generalized to distributed governance but instead localized to education specifically.

4.1 Consensus - Proof of Affordable Education

Analogous to bitcoin's proof of work scheme (PoW), proof of affordable education (PoAE) will be the burden of proof placed upon educators to receive financial incentives. Similar to the work done by backfeed [cite backfeed paper], blockEDU will use a distributed reputation system to provide proof to the Ethereum smart contracts that courses were taught in a valuable way.

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. A monetary reputation is assigned to the educator as a separate unit to decide that it was done for free by the students. This of course can not be enforced in any digital protocol as teachers are free to charge in any fiat currency they like outside of the system. However students do have a course of action to ensure that teachers are not paid twice in EDU by lowering the PoEA reputation at the end of the course. A student may get a good education from a teacher, but still assess a false monetary flag for the course. If either of those conditions is broken, the teacher will not be paid out from the reserve token.

A special choice is uploaded to the course contracts once completion of the course has happened. The choice is whether or not to reward the educator for mining the course. This gives students 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. This decision is uploaded by the context area agent for that course and determined locally by assessing the recommendations given by the students in the reputation system and assigned as a binary value [0,1] upon course completion. This binary decision is not recorded by the agent in the reputation system but it is derived from it.

4.2 EDU Tokens

EDU represents the unit of consensus that education should be affordable. EDU is an ERC20 standard token. The total supply is held in reserve on the Ethereum blockchain. The tokens provide a method for rewarding the educators after completing the proof scheme and maintaining the consensus that education should be affordable. The tokens will work as a method for students to provide a fee for entering the courses. These tokens also provide a way to unlock the data storage needed to hold course material and student records permanently and securely.

4.2 (a) zEDU Tokens

To continue with the objective that any users in the protocol may remain private, we offer a mechanism for converting traceable Ethereum tokens to [ZKSNARK] protected 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. This is done by [cite sapling, read method]... This will give private payment methods to both students and educators in the event that certain education becomes oppressed or deemed illegal by authorities.

4.2 (a) Mining Tokens

While there is no computational methods of mining or stake to ensure the network reaches consensus, a loose analogy to bitcoins [site 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 score of the course passed in by students.

4.2 (b) EDU Blocks

A course in the protocol is like a block. This is similar to bitcoin mining where miners create blocks of transactions to be rewarded but instead teachers are creating courses filled with 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 the course duration runs out as their mechanism of work.

4.2 (d) Token Fees

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 for entering the course. This is analogous to a bitcoin transaction fee, where a miner is rewarded an extra amount for picking up this 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 (e) Reserves

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

4.2 (e) Block Rewards

An issuer contract holds the reserve of all tokens that will be issued to educators. When a course contract is completed, 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 will have TTL for the course length as provided by the educator setting the course up via a CMS. Each course contract will have four states that will open their contract up rewarding the educator from the issuer contract. This way a teacher may be paid in installments during the length of the course.

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 lose their reputation when other nodes notice this and be unable to facilitate further reputation and report to the contracts as designed by the UniTEC system.

Every quarter the course contract will allow for the nodes to begin reporting the reputation to the contract. The fields that are able to be reported on are an *affordability value* and an *education quality value*. Recommendations are issued by the students In order for a node to know how to report correctly to a contract during each quarter of the courses TTL. The affordability and education quality values are calculated between [0,1]. The delta threshold 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 not a static value. Courses should have different values depending on how reputable the educator giving the course is. In this way a teach with a high value of recommendations is paid more for their time. Teacher recommendations come with a value based score for each 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 a block reward they should get. If Alice has a teacher value score of 0.8, then she will be rewarded 80% of the highest block reward possible during each of the quarterly payout rounds.

In order for the contract to get an accurate report on the reputation and affordability score for the educator it must not take the first report it sees. There will be a window of time that each contract can take scores for each payout period. Since not all nodes in the reputation system will 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 will come with the 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 node that reports with the highest number of seen recommendations.

5 Distributed Reputation System

The distributed reputation system deployed in the blockEDU protocol is based on the research done in the UniTEC system [cite UniTEC]. The reputation system plays an important role in the blockEDU protocol and is important to understand how UniTEC works. Section 6 is mostly an outline of the UniTEC protocol with modifications to reflect that blockEDU has a blockchain to work with and is scoped to the reputation of education. We try to take a general protocol and show how it will be used in the education context.

5.1 Overview

The UniTEC reputation system paper is summarized here, pulling out what we believe to be the core elements to understanding such a complicated distributed reputation system. The UniTEC

authors put a great deal of diligence into covering much more information than is outlined here. It is encourage that the reader review the UniTEC design for a better understanding of the reputation system deployed by the blockEDU protocol.

Our variant of the UniTEC reputation system works by allowing users to interact with nodes on the network to decide who to trust. Nodes form trust in other nodes and thus should be required to behave properly, or they would not become a part of the trusted network. This network is responsible for determining whether a teacher can provide reliable education to their students, determining if students can trust other students to be good peers in a course, determining whether the course offered was of value to the students and education was passed to them, whether or not the course was provided affordably, and in general whether a node can trust the recommendations passed to them from another node.

Once we have a network of trust built between distributed nodes, we can leverage this network to provide the quality feedback that the Ethereum contracts needs to enforce the protocol. This network can be view as a distributed oracle providing the data needed to maintain our consensus protocol.

5.1 (a) Identity

Identity in our system is in the same format as Ethereum [source the format, 160 bits, encoding details]. 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 Orc Network and Zcash tokens (should privacy in token be required). These identities work the same way with slightly different formats [discuss orc / zcash format].

As to not burden each user of the system and fulfill the requirement that the protocol is as easy to use and easier to set up than 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 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.

5.1 (b) Full 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 may hold the identity private keys and sign for

users], they report as oracles to the courses that lie within their context area, and find trust paths so that the ease of use requirement can be fulfilled.

Each node/s is responsible for managing a context area 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. As they are responsible for connecting each user to the reputation system, and the reputation system is the proof scheme responsible for maintaining consensus, it is essential that these nodes be trusted. Just as every identity in the system will have a reputation, these nodes will have a reputation and this reputation will be used by users when selecting a node/s to service them. Any user may decide to run a full node and advertise themselves as a facilitator. Ideally the entire network would be made up of these full nodes and not end users.

5.1 (c) Node Fees

A secondary EDU mining method is created here to incentivize these nodes to facilitate users who do not have the technical ability to be one. This is important due to the cost that these nodes incur for satisfying this role. The nodes must pay for the farmers in the Orc network to hold the recommendation data and facilitate the bandwidth traffic necessary for the communication between other nodes. They also must pay miners in the Ethereum and Zcash network for interacting with the contracts.

An optional request fee in EDU will be given to each request a user makes to a full node. Each node may choose to facilitate requests if there is no fee, or they may reject the request and require a fee from the user to cover their costs or make a profit.

5.1 (d) Offline Nodes

Since a node is the bridge for users to access the network, it is important that there are backups when a node goes offline. Pointers to the location of 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 backup in the event that the nodes facilitating their identities disapear.

5.2 Modeling

Below we describe the models of the distributed reputation system as formed by UniTEC, the *trust*, *knowledge*, and *system* models.

5.2 (a) Trust Model

Trust Context Areas

As the UniTEC paper describes, trust is not all-encompassing. Instead people's trust is relevant to a certain context. Someone who is trusted to repair your car should not have this trust translated over to another context area of say babysitting your children. The context areas in the blockEDU protocol are partitioned by knowledge areas i.e. Mathematics, Physics, or Literature. We also have to superset of context areas to differentiate students and teachers. FInally we have the lowest subset of areas for each individual course that will record the scores necessary to discern if the course was provided with quality and with affordability.

Here we must make a hard partition between the reputation that is derived for the student/teacher context area and the course field (Mathematics/Literature, etc) from the individual course reputations. We do not want recommendations from students that have not been enrolled in a certain course carrying any trust confidence weights over that specific course. In this way we can prevent students from continuing to rate courses that they are no longer enrolled in as each course should be independent from all others. The overall rating of the educator in the superset of educators does carry over between courses so the issuer contract knows how much to reward each teacher. This is to avoid the case that a teacher could collude with one student, teach a course of little value to that one student, and be rewarded the same as a teacher providing a valuable service to many students.

Each context area has a trust value in the range from [0,1] and a given confidence vector in that user for giving the recommendation of that value. In this representation, 0 indicates that there is either no previous experiences with the user in the given context area while a 1 represents maximum trust in the user for either giving recommendations in teaching a course, being an experienced student, or being a reputable student that has the ability to confirm the consensus proof. I may trust Alice with a value of 1 meaning that I believe she is a capable recommending a good English teacher.

The confidence vectors store metadata that defines the quality of trust in the entity and contains the following entries:

- Number of direct experiences in that context area
- Number of indirect experiences (from context areas influencing the one in question)
- The last N direct experiences
- A blacklist (set by the user to prevent recommendations from certain identities)

Keeping track of the number of experiences with another user's gives the system a way to calculate trust on the fly. It is a parameter to the trust update algorithm as discussed further in section **5.6**.

5.2 (b) Knowledge Model

The UniTEC system develops the knowledge model as a "local profile stored for each user to create a view of "who knows what" including the user's own knowledge. Hence it is used to communicate one's own expertise as well as to learn about other's expertise."

The cardinality of the set of personal recommendations is stored to build the *own expertise* indicator for a certain context area. Figure 6.2 shows an example of how these recommendations are stored to build this expertise. We do not currently explore pseudonyms for each participant where a user could have a different name for each context area they have knowledge in, but this has been done in the original UniTEC design.

To evaluate the *expertise of others*, we store the number of recommendation requests a user gets for each context area. This is also how we express each user's understanding of a context area.

[figure 5.2 Sample Knowledge model entry]

Definition (Authority): *Someone that is expected to have a direct knowledge of the given context area and can provide helpful recommendation in that area.*

Definition (Hub): *Someone that does not have a high level expertise in any one particular context area, but knows many Authorities that do.*

Recording both of these pieces of information allows us to to discern information about each user's expertise as an authority (likely a teacher or advanced student themselves) or a hub (perhaps a dean of a university or a teacher's assistant).

5.2 (c) System Model

Our system will be comprised of both light clients (end users) and full nodes. Nodes will be responsible for storing both the trust and knowledge models described in section **5.2 (a)** and **5.2 (b)**.

[create a graphic here showing the two top level context areas (student and teachers) as a well as the subsets for each (the different academic areas of interest). This will give an idea of how neighborhoods are formed and how trust will be built by what is relevant to each individual user]

The systems develops neighborhoods based on context areas. When a user wishes to be informed of a certain trust in another user, they reach out to the network and look for a connection in their neighborhood by issuing recommendation requests. Hops in the network are described by whether or not a facilitating node is receiving direct recommendations from another node that it is connected to and trusts. For example, if Alice wished to take a Mathematics course from Bob, she may begin a request for a recommendation from Bob by first contacting a node or nodes in the educators context area superset in the neighborhood of Mathematics. UniTEC defines neighborhoods as follows:

Definition (Neighborhood): *For a node N and a trust context area C, the neighborhood ONet(context area C, node N, level L) is the set of identities that can be reached from N in L hops. The set ONet(C, N, 1) is the set of identities directly connected to node N. The membership of an identity I in this local view or level 1 view of the neighborhood of a node N is determined by an algorithm that relies on two inputs: N's trust in I (stored in its trust model) and I's advertised expertise (from N's knowledge model entry about I) Generally:*

$$ONet(C,E,L) = \bigcup_{\forall E' \in ONet(C,E,1)} ONet(C,E',L-1)$$

[consider giving the example in UniTEC paper]

5.2 (d) Recommendations (RECs)

UniTEC provides a data structure for storing trusted data items that serve as recommendations that we are calling RECs. RECs in blockEDU are comprised of *recommendation data*, *recommender data*, and *metadata*.

Just as in UniTEC we attach RECs to a specific context area and allow for these units to be flexible and contain arbitrary fields that specify the recommendation target and content. blockEDU recommendation data is as follow:

```
* @value {Boolean} Recommended - Positive or negative REC
* @value {float} quality percent - The object has a 65% quality rating
* @value {String} arbitrary length - Text specification of the REC
```

The recommender data contains the ethereum pubkey of the recommender and their confidence in the given recommendation. This gives a statement about the recommenders own confidence in their recommendation and influences the trust update when processing the requester's feedback

```
* @value {String} Identity - Ethereum Hex encoded pubkey identifier
* @vaule {Integer} Recommender Confidence - between 0-10
```

The metadata contains a timestamp and the recommendation ID. This is to give each recommendation a TTL that affects the ratings in the trust algorithm. Optionally there is a blacklist that can block identities from receiving the recommendation.

```
* @value {Unix Epoch} Timestamp - A TTL for RECs
* @value {Array} Blacklist - List of identities to not forward to
```

Recommendations are signed by the ethereum private key of the recommender in order to prove the authenticity of the REC. We now have a system that supports a REC data structure that can be verified and quality checked.

5.3 Interactions

Here we describe how the models and components discussed until this point will interact in relation to how the reputation system publishes and locates recommendations. The process as described by UniTEC is divided into five subtasks, "*disseminating* the request, *collecting* the responses, *organizing* the results, and *providing* and *processing* feedback for the system".

5.3 (a) Publishing Recommendations / Advertising Knowledge

Any user may publish RECs as a way to provide experiences of a student teacher, student student, or teacher teacher relationship. For the student teacher relation, these RECs provide a way for students to get a good idea of whether or not an educator has good past experience with their students and is knowledgeable about the given field of study. Students may want to read experiences with other students to see if they would be likely to have a good experience with others in a course they are about to take. Teachers may publish their experiences with other teachers in the field which can also establish a greater trust that they are qualified to give the course.

The other side of publishing RECs comes into the system when a course completes and the consensus protocol needs to know whether or not to **a)** Decide to take this REC into account. If the student is not enrolled in the course then it should be discarded by the node processing it. **b)** Decide if the course received a good rating from its students and is within the delta for the monetary rating suggesting it was provided affordably, then the teacher should be rewarded the mining fees and block reward for the course.

When users fill out the REC fields and want to publish, they may send the REC to their designated node/s. The node will sign with the associate ethereum private key and timestamp the REC. The REC is stored in the nodes Orc bucket and the node's REC pointer is updated in the Ethereum recommendations registry. The knowledge entry for the node is increased as it now has a new REC. To avoid excess Ethereum costs, these pointers do not have to be updated for every REC received, just when there is a threshold that is hit that is determined safe enough to probabilistically not lose data.

In order for a user to request RECs, their node must gather knowledge from other "identities with respect to stored RECs per context area. This issue can be addressed via two different approaches: *advertisement messages* and *knowledge providers*". When the threshold for RECs for a certain identity is hit, an advertisement message will be sent on the pubsub system with the number of RECs that identity has for each of the context areas they are associated with.

Those receiving these advertisements will be other nodes that are subscribed to the context areas the RECs are in as well as the knowledge provider **[contract?]**.

The knowledge provider contract is a registry that stores the expertise areas of identities. As Alice begins rating her professor Bob in mathematics, the knowledge contract will be updated with a record that her identity is experienced in Bob's courses. This helps prevent the start up problem of their not being enough direct experts to find RECs about Bob's course when there has not been many student that have taken the course. At anytime a node may consult the knowledge contract by passing Bob's public key to see who knows about him.

5.3 (b) Handling Requests

As UniTEC defines, the request message contains the *identity* of the requester, a *request identifie*r, the *recommendation target*, the *trust chain* and a *hop counter*. The hop counter and the number of identities that are reachable define the number of identities the requests reaches.

The *trust chain* is built during the request and gives a confidence to the requestor that they can trust the recommendation they are being served. If Alice wishes to get a recommendation for Bob who is three intermediaries away, the full list of intermediaries and their corresponding trusts will be sent back to Alice so that she may decide whether or not Bob is for example, a good teacher, or whether Bob's course was provided affordably as recommended by the students in the course in the event that a node is requesting to report. Alice's request is first sent to her node, which forwards the request to all reachable members in the level 1 neighborhood of the request context area. Each recipient of the request for L hops computes the algorithm described in figure **5.3 (b)** as first put forth by the UniTEC system. Each new request created by the identities in the neighborhood of the context area of the request is formed by creating an addition to the signed trust chain link they are adding with their level of trust in the next hop to the recommender. The identity of the person they are passing the request to is also attached. Section **5.7** goes into further detail about how these chains are used to derive trust in the requested identity from the requestor, as there may be more than one path to a requested identity.

[create graphic to illustrate a request]

5.3 (c) Collecting Responses

Those nodes that are able to fulfil the recommendation for an identity that was requested (meaning they have a stored REC for the corresponding requested identity) will create a recommendation response for the requestor. The response contains a request identifier, the signed REC, and the trust chain with the last link being the trust of the identity in the recommender.

```
calculate trust of Req. in Ii via the current trust chain
if not((request already processed) AND (with higher trust)) {
   if (suitable recommendation available) {
      create recommendation response
      send recommendation response back to Req.
   }
   decrease hopcounter
   if (hopcounter > 0) {
      foreach (member Ij of ONet-1 of Ii) {
         (* create set of new requests *)
         create copy of received request
         insert digitally signed trust statement:
         {
            trust of Ii in Ij
            (pseudonymous) identity of Ii
         }
         send request to Ij
      }
   }
}
```

Figure 6.3 (b) UniTEC simplified directed dissemination algorithm

Only the most recent recommendations decided from the timestamp are kept. The requestor may receive multiple recommendations for the identity in the request and the ones with the highest trust ratings are displayed first.

5.3 (d) Handling Feedback

The requestor needs to make a statement about which recommendations received were perceived as useful. The UniTEC system describes three steps for handling feedback by *collecting the user feedback*, *updating the trust* and *updating the neighborhood*. This will require interaction from the user but it creates *experience* with the recommending identity which is either positive or negative. “For each of the identities the experience and recommender confidence are added to the trust model and the number of direct experiences is increased in the appropriate context area”.

This feedback described in the previous step is used in what is called the trust update algorithm. Section **5.6** goes into further detail about how blockEDU uses the original trust update algorithm presented by UniTEC. It is noted that algorithms are components that can be swapped in our reputation system if it decided that a better one suites the blockEDU protocol.

The level 1 neighborhood is then updated to reflect the new trust values. UniTEC cites Claus Offe who rationalized that it would be inappropriate to only get trust values from already trusted identities thereby abandoning the possibility for gaining experience from other sources. Choosing the neighborhood comes from the highly trusted reputable identifies the node knows about, consulting the knowledge contract for knowledgeable identities, and selecting random identities in the network from the DHT.

5.4 System Architecture

The UniTEC research uses a Coral [cite coral] DHT as a method for distributed communicating and storing the reputation information that has been laid out previously. blockEDU uses a pubsub DHT for communicating, however we diverge from UniTEC here by storing reputation data directly in the Orc Network. [this needs to be discussed, it could be too slow to combine two networks in this request as users are looking through the blockEDU dht and making requests to the Orc nodes each time it needs rep data. Using a coral DHT and storing the meta information may be a better approach].

5.4 (a) Data Management Component (DMC)

A node's data storage happens both locally [(Orc?)] and remotely (DHT). The local storage stores information from the trust and knowledge models described in section **6.2** as well as the RECs created by the user. An additional limited number of RECs are stored that have accumulated from requests from users. The RECs are formatted in JSON and transmitted over JSON-RPC. Figure **5.4 (a)** shows an example of a REC.

Figure 5.4 [create an example JSON REC]

5.4 (b) Peer-to-Peer Overlay Component (POC)

The POC is responsible for managing the neighborhood view each node has. This is notified by the trust management component (TMC) upon receiving knowledge advertisements, RECs, and feedback on the quality of RECs. When a node receives a request for a certain recommendation, the POC checks the following:

Check 1: Check the trust chain in the request and check if the last link points to an identity that the node knows about. The request is discarded if not.

Check 2: Validate the signatures on all links in the trust chain and discard the request if any of them are broken.

Check 3: Contact the TMC to calculate the transitive trust of the requester in the current local identity. If the request has not been processed before (check by the request identifier) or if it has been processed with a lower trust value, the new trust is stored and processing continues else it is discarded.

If a REC is found that satisfies the request then it sent back directly to the originator of the request with a finalized trust chain. RECs that are found that originate from non local identities to the node then multiple messages are sent back for each for each non-local identity with an expanded trust chain containing the trust of the local identity in the respective non-local identity.

Finally if the hop counter in the request is not 0, the request is disseminated to the neighborhood. Each request sent to a neighborhood member contains a trust chain that is expanded with a link containing the trust of the local user in that particular member.

5.4 (c) Trust Management Component (TMC)

The TMC handles evaluating the user's transitive trust in the REC issuers after receipt of the REC responses, updating the user's trust in the REC issuers after the feedback step, and handling and updating the expertise information. Multiple trust chains are evaluated here which is discussed further in section **5.6.**

The TMC keeps track of the trust in each identity it has been in contact with, specifically storing the trust in the DMC database in accordance to the trust model. The TMC updates the trust in the identities when it receives feedback on the quality of received RECs according to the trust update algorithm discussed in **5.6** which influences the neighborhood selection the next time a query is received.

"Updating the expertise of identities in the knowledge model is independent of user feedback. The authority rating of the received RECs recommender is increased by 1. The hub rating of all intermediaries in the trust chain(s) of each REC is increased by an amount corresponding to the distance of the intermediary to the recommender. This approach ensures that intermediaries connected to many recommenders eventually appear in the immediate level 1 neighborhood of an entity for a certain context area".

5.5 Trust Updating

Trust update algorithms in general specify how to compute trust from a certain given set of inputs. Due to the complexity of deriving meaning in trust from events, there are different algorithms that may be well suited for different users. This section highlights the chosen general trust model as described by UniTEC. This contribution by UniTEC shows that, while algorithms to compute trust values are different, they rely on the same data and often come to the same conclusions. With this generic model, we may have the flexibility to understand multiple update algorithms, being able to fine tune the algorithms in blockEDU protocol to best suit the needs of its users.

Reputation of an identity is the average trust the network has for the entity, where trust is derived locally from experiences. There is a global quality to reputation while trust is derived subjectively in the given context area. Here we begin setting up the understanding of a generic trust model.

5.5 (a) UniTEC Trust Relationships

Here we define further what trust means in the context of our reputation system as first described by the work done by UniTEC.

Trust measure: This is the measure of quality of the trust relationship ranging from distrust to full trust.

Trust certainty: Based on personal experience, this the measure of confidence in the trustee by the trustor

Trust context: This is defined in section **5.2 (a)** as the context areas.

Trust directness: This specifies between direct and referred trust. If Alice has a personal experience with Bob, this would be a direct trust. While if Alice was told to trust Bob by Clark, this is referred trust.

Trust dynamics: This defines the way in which trust is updated over time.

5.5 (b) Generic Trust Model

The trust measure selected by UniTEC and subsequently blockEDU protocol is defined in the interval [0,1]. Complete distrust is represented by 0 while complete trust is represented with 1. Trust certainty is likewise measured in the same interval.

Context areas are not necessarily independent of each other. For example the fields Mathematics, Computer Science and Physics share many of the same concepts. It can be seen that physics and computer science have an *part-of* relationship with mathematics. This measure may not work for all areas so we formalize a distance measure between context areas in the interval [0,1]. A distance close to 1 represents a high dependency, for example the distance between math and computer science may be a 0.6 while the distance between math and physics may closer with a 0.7. Do to the subjective nature of trust, these distances may be modified by each user to accommodate their own personal views. Doing this allows us to spread the impact of trust updates in one area throughout the whole reputation system. This semantic can however not carry over to our trust in the consensus algorithm, where each course is localized to their context area island. We want a strict definition of trust in regards to whether or not a given course was offered affordably.

5.6 Trust Transitivity

This section describes how trust is derived in the trust chain. Trust transitivity means that if Alice trusts Bob, who in turn trusts Clark, then Alice will also trust Clark.

5.7 Defending Against 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.

6 Distributed Cloud Storage - Orc Network

[cite Orc]

A persistent data storage network is needed for both educators and students to store course content and reputation records. The Orc Network offers [Have at it Gordon :)]

7 Ethereum Contracts

Ethereum is a decentralized network of nodes that process transactions and maintains consensus on general state. [give better def and cite Ethereum]. Ethereum is used in the blockEDU protocol to maintain the state of reward system in place for educators. It will provide a mechanism for unlocking courses / course content, it maintains the identities of the users in the system, it provides the registries that provide pointers to the current state of the applications running on the protocol.

7.1 Ethereum Shortcomings

Storing data in the ethereum network is expensive. [gather details on op storage costs per byte of hex data]. Ethereum is used as the network to provide a decentralized point of authority on the state of the blockEDU protocol. 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 channel research here]

7.1 b Linked Data pointers

The Orc Network provides 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.

The meta information needed to locate a file on the Orc Network is as follows.

```
* @param {String} id - Unique bucket ID
* @param {String} file - Unique file ID
```

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. The total cost for a X byte MLOAD instruction is X units of gas. [fill in the Xs]

```
pragma solidity ^0.4.11;

contract OrcDataPointer {

    address owner;
    bytes32 pointer;

    modifier onlyOwner() {
        if (msg.sender != owner) throw;
        _;
    }

    /// Create ownership of this pointer to data.
    function OrcDataPointer() {
        owner = msg.sender;
    }

    /// resets the pointer.
    function removePointer(address to) onlyOwner {
        pointer = 0x0;
    }

    /// Update the given pointer to a new state of the data.
    function updatePointer(bytes32 _pointer) onlyOwner{
        assembly {
            pointer := mload(add(_pointer, 24))
        }
    }
}
```

Figure 7.1 Simple contract to store pointers to the orc network.

As shown in Figure **7.1** a simple method in a contract can be used to load pointers to Orc data onto 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. It should be noted that the above example does not give an accessor function to the loaded data.

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.

7.3 Course Contracts

7.4 Registry Contracts

7.4 (a) REC Registry

The REC registry stores the pointers for each full node's distributed database storing the recommendation data for its users. This registry is updated everytime a new recommendation in any context area is published to the node facilitating the request to publish,

7.4 (b) Knowledge Provider Registry

8 [create a name] - The first CMS

The first CMS is built by blockEDU on-top of the protocol token, reputation system, and Orc 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 Orc Network distributed storage method presented here. [enter name] serves as an example of this may be done.

[description of the CMS we are building, similar to udemy]

8.1 Client Application Features and Scope

8.2 Requirements Analysis