УДК 004.7

ВОЗМОЖНАЯ АРХИТЕКТУРА СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ ДОСТИЖЕНИЙ СТУДЕНТА УНИВЕРСИТЕТА НА ОСНОВЕ ТЕХНОЛОГИИ BLOCKCHAIN

А.И. Мартышкин
кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье проводится результат исследования применимости технологии Blockchain для реализации распределенных реестров личных данных, на примере реестра достижений студента университета. Рассматривается организация взаимодействия в системе. Приводится выражение для расчета Хэша транзакции.

Ключевые слова: распределенные системы, блокчейн, криптовалюта, информационные технологии, реестр, токен, хэш-идентификатор.

POSSIBLE ARCHITECTURE OF THE STORAGE SYSTEM OF DATA OF THE ACHIEVEMENTS OF THE STUDENT OF THE UNIVERSITY BASED ON BLOCKCHAIN TECHNOLOGY

A.I. Martyshkin
Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The article presents the result of a study of the applicability of the Blockchain technology for the implementation of distributed registries of personal data, based on the example of the university student achievement register. The organization of interaction in the system is considered. An expression is provided to calculate the transaction hash.

Keywords: distributed systems, blockchain, cryptocurrency, information technology, registry, token, hash identifier.

Введение. Внимание к технологии Blockchain привлекла возросшая популярность основанных на ней криптовалют. В 2009 году Сатоси Накамото опубликовал исходный код биткоина.

Цель работы. Цель работы заключается в исследовании применимости технологии Blockchain для реализации распределенных реестров личных данных, на примере реестра достижений студента университета.

Материал и результаты исследований. Реестр (хранилище) данных в данной работе будет представлять собой «визуальное резюме». В нем будет отражаться уровень квалификации, опыта и профессиональных качеств его обладателя. Для реализации будем использовать зачётную книжку. Это

документ, содержащий записи об успехах студента – сдаче зачётов и экзаменов, прохождения различных практик, а также отметок о сдаче курсовых, ВКР, диплома или диссертаций. Далее рассмотрим состав реестра достижений студента.

Для реализации заданного реестра рассмотрим компоненты, необходимые для создания системы хранения и подтверждения подлинности достижений студентов университета. Это модули межсетевого взаимодействия, модуль работы с записями (вычисление ключей, сверка данных и пр.), модуль работы с базой данных. В ходе работы реализуется два приложения. Первое – для пользователей, с помощью которого будут добавляться транзакции, а также просматривать данные. Второе приложение – валидатор, для него должен предоставляться доступ к базе данных университета.

Модуль межсетевого взаимодействия проектируется на основе технологии peer-to-peer. Технология P2P отличается от стандартных методов развертки сетевых инфраструктур. Когда применяется подход "равный-равный" то, коммуникация осуществляется не по пути клиент-сервер, а по пути нахождения других клиентов в сети, в этом случае они могут обмениваться данными между собой напрямую [1].

Возьмем в качестве примера для взаимодействия сайт www.penzgtu.ru. Приемная комиссия университета выложили список поступивших на бюджет. Соответственно, накануне размещения списка сайт практически в одно время посетит большое число абитуриентов. Как только список будет опубликован, будущие студенты в этот момент будут загружать его и есть вероятность, что сервер университета не выдержит нагрузки и «ляжет». Чтобы не допустить такой ситуации, оптимально было бы применить технологию peer-to-peer. Файл вместо загрузки каждым участником с сервера, сначала рассылается небольшому числу пользователей, а остальные пользователи будут загружать его от тех, у кого он уже есть и так далее. Этот процесс можно дополнительно ускорить, если разбить файл на мелкие части и разные части файла скачивать из разных источников. По похожей технологии работают Torrent-системы [1].

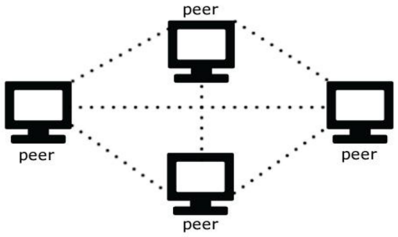

Рисунок 1 – Peer-to-Peer

Каждый участник сети должен иметь возможность выполнять такие функции, как нахождение других пиров; соединение с другими пирами; обмен данными с другими пирами.

От модуля работы с записями требуется выполнять вычисления устойчивых ключей на основе SHA-256. Хэш-функции семейства SHA-256 вычисляются на базе структуры Меркла-Дамгарда. Исходное сообщение делится на блоки, один блок – 16 слов. Структура пропускает блоки через цикл с 64 или 80 итерациями. В ходе итерации 2 слова преобразуются, функцию преобразования задается остальными словами. Затем результаты вычисления каждого блока суммируются, сумма и будет значением хэш-функции.

Модуль работы с базой данных должен иметь возможность проводить стандартные операции с базой данных и используя модуль межсетевого взаимодействия сообщать другим пользователям сети об изменения в базе данных. В качестве СУБД будем использовать MySQL

Рисунок 2 – Схема архитектуры системы

Рассмотрим подробнее организацию взаимодействия в приложении на диаграмме последовательности, которая представлена на рисунке 3.

Допустим, участник сети решил добавить запись (рисунок 4). Для этого он заполняет необходимые поля: сетевой адрес студента, сетевой адрес преподавателя, наименование предмета, вид аттестации и оценка.

Сетевой адрес студента формируется так. При регистрации в сети блокчейн создается пара закрытый - открытый ключ. Закрытый ключ знает лишь владелец, а открытый может размещаться в сети. По открытому ключу и выполняется хэширование. Затем результат хэширования ещё раз хэшируется и преобразуется в Base58. Это и есть сетевой адрес [2]. После заполнения всех полей, для проверки подлинности записи, отправляется сообщение на узлы валидатора, находящиеся в сети (рисунок 5).

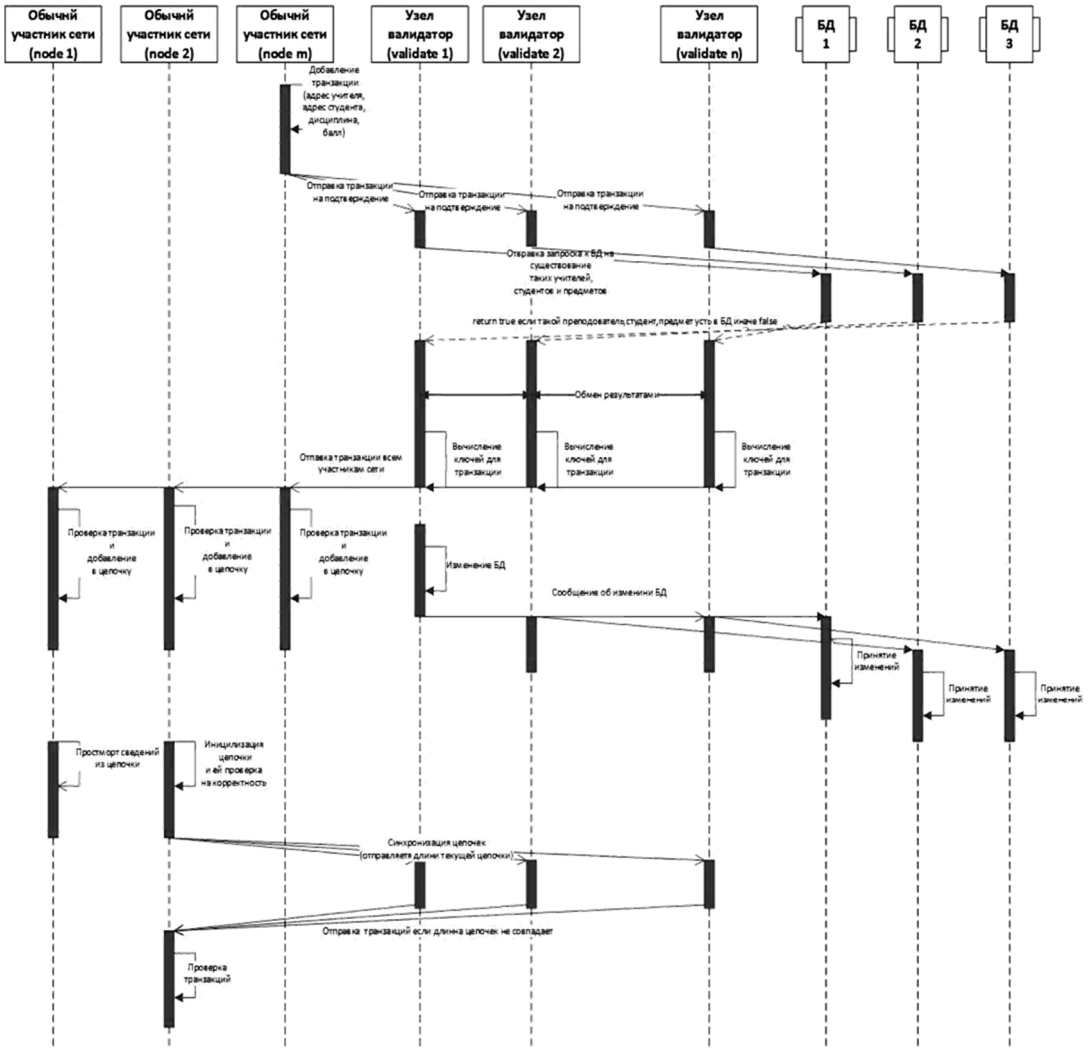

Рисунок 3 – Диаграмма последовательности

Рисунок 4 – Диаграмма добавления записи

Узлам-валидаторам необходимо наличие доступа к базе данных с информацией о студентах университета, а также преподавателях, для возможности проверки записи, то что существует студент, которому выставляется оценка, существует этот преподаватель в университете, есть такой предмет

и что этот преподаватель может выставлять оценки по этому предмету. Процесс изображен на рисунке 6.

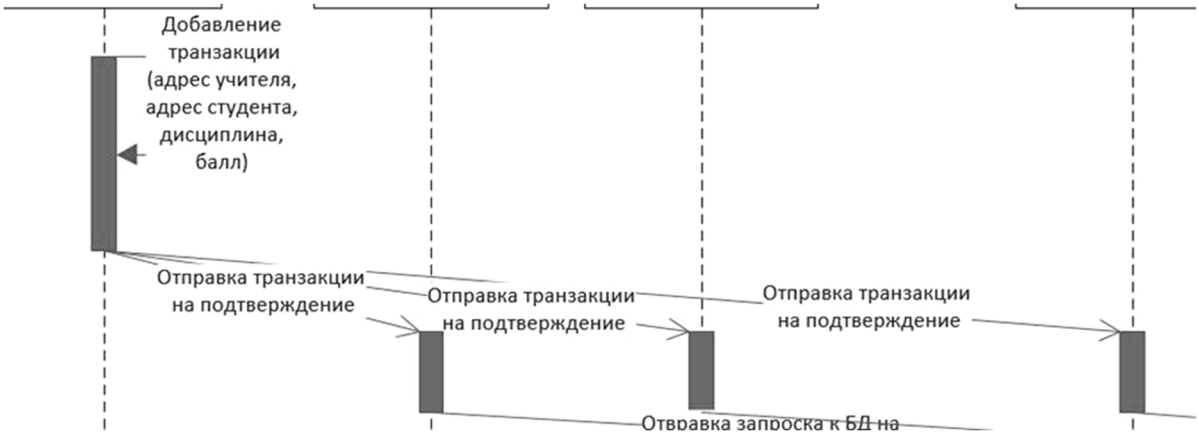

Рисунок 5 – Диаграмма отправки записи

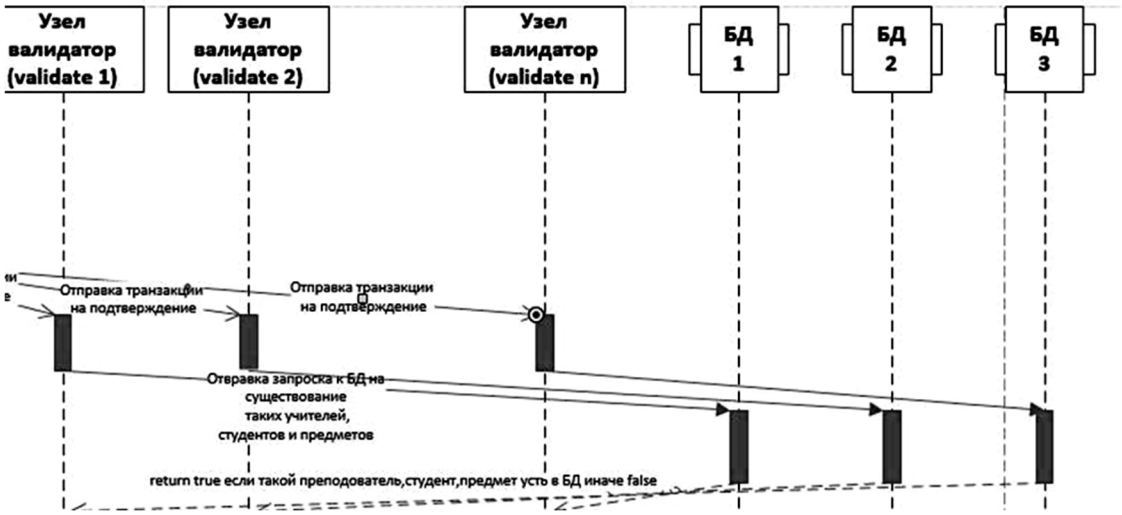

Рисунок 6 – Диаграмма последовательности взаимодействия с базой данных

После проведения проверки, все валидаторы передают друг другу сообщения с результатами. Таким образом у каждого валидатора есть набор данных о результатах всех проверок, и если большая часть валидаторов подтвердит законность транзакции, то начинается процесс расчета ключей (рисунок 7).

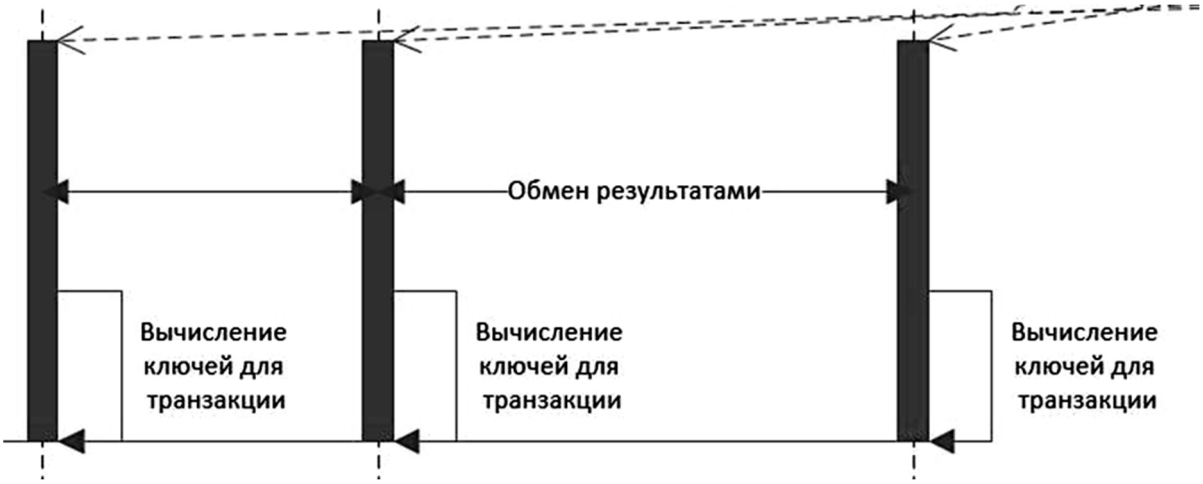

Рисунок 7 – Диаграмма последовательности подтверждения законности записи

Запись, или транзакция имеет следующее представление (рисунок 8).

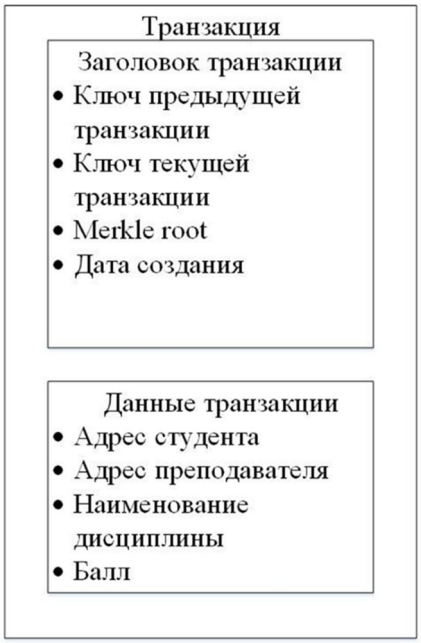

Рисунок 8 – Схема транзакции

Хэш транзакции высчитывается согласно выражению

$$H_{i+1} = SHA-256(SHA-256(M_{i+1}) + SHA-256(H_i)), \quad (1)$$

где Hi+1 – текущая транзакция, Mi+1 –дерево Меркла для текущей транзакции.

Дерево Меркла – один из видов структуры данных, также известная как бинарное дерево [3]. Для исследуемой системы оно строится таким образом: на первом этапе рассчитывается хэш от дисциплины. Далее рассчитывается хэш количества баллов. Затем рассчитывается хэш от суммы хэшей предыдущих пунктов. Полученное значение называется кумулятивный хэш (merkle_root).

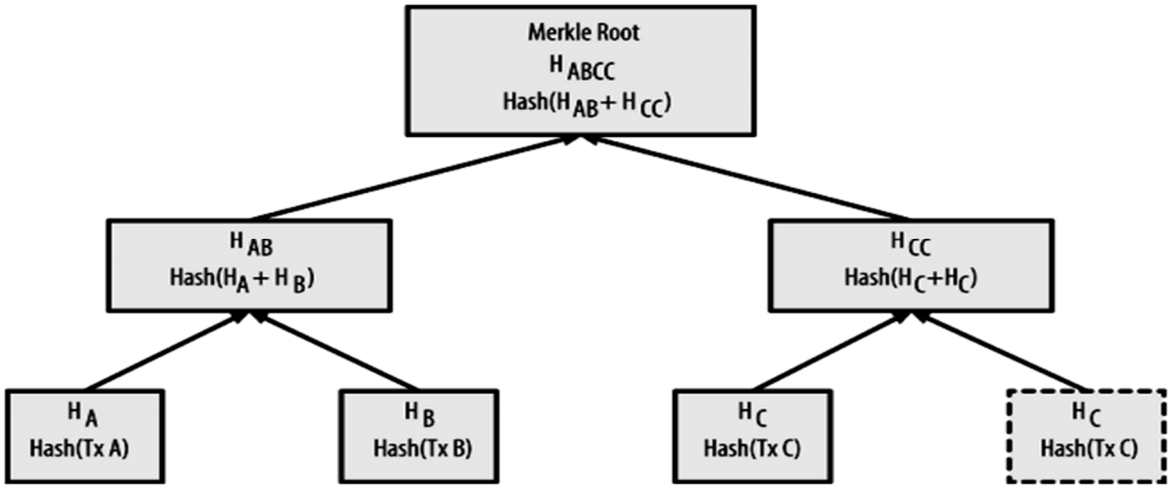

Рисунок 9 – Кумулятивный хэш

Необходимо уточнить, во-первых, зачем эти усложнения и во-вторых, почему мы не используем один валидатор. Так как технология blockchain достаточно новая, требуется её постепенная интеграция в имеющиеся системы хранения данных. По этой причине мы будем использовать компромисс между blockchain и традиционными системами. Поэтому узел валидатору необходимо наличие доступа к БД. А наличие нескольких узлов валидаторов требуется для выполнения условия децентрализации, такая модель имеет более высокую защиту от взлома и подделки данных.

Для просмотра данных о любом студенте, достаточно увидеть имеющеюся цепочку по его сетевому адресу. При запуске программы происходит инициализация цепочки из файла. json. В этот момент идет проверка на корректность данных (не вносились ли изменения в файл). Если проверка не пройдена, то цепочка остаётся пустой. После инициализации системы происходит синхронизация цепочек с другими узлами сети. Текущая длина цепочки отправляется на узлы валидаторы. Когда длина цепочки узла короче, чем у валидатора, то недостающие записи подгружаются на узел, после чего система готова к дальнейшей работе (рисунок 10).

Вывод. следуя вышесказанному, сделаем вывод, что при помощи технологии blockchain возможно построить систему хранения, либо реестр хранения данных.

Работа выполнена при финансовой поддержке гранта РФФИ «Конкурс на лучшие проекты фундаментальных научных исследований» (Грант № 19-07-00516 А).

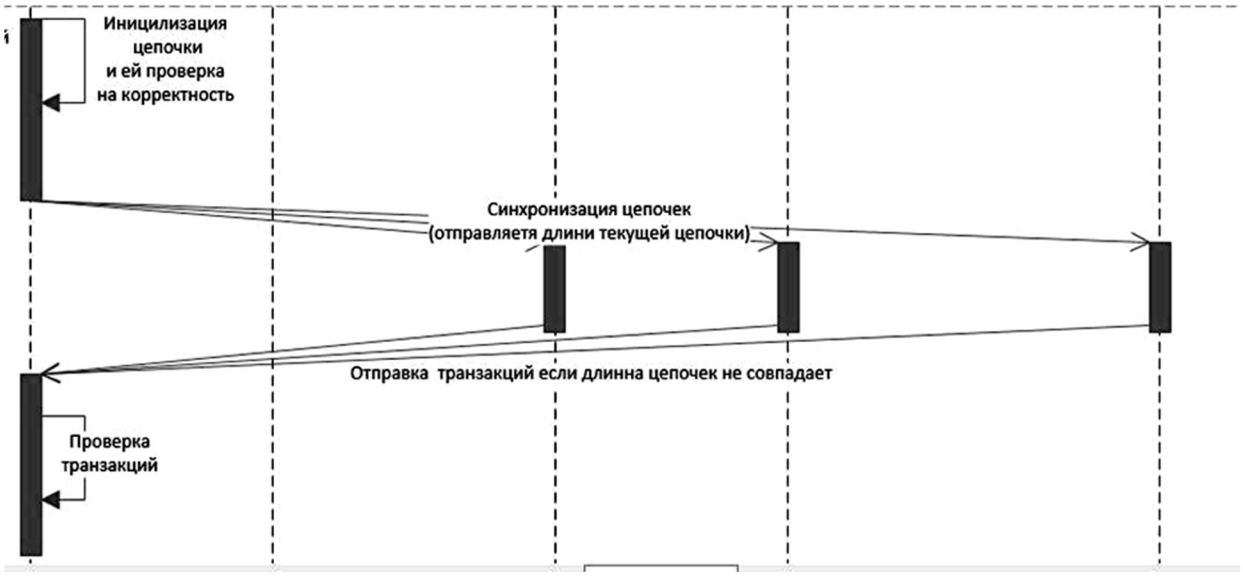

Рисунок 10 – Диаграмма последовательности синхронизация цепочек

ЛИТЕРАТУРА

1. SHA-256 [Электронный ресурс]. URL: http://www.iwar.org.uk/comsec/resources/ cipher/sha256-384-512.pdf (дата обращения: 28.03.2019).

2. Proof of work. [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/ Доказательство_выполнения_работы (дата обращения: 25.03.2019).

3. «Токены vs Пароли» [Электронный ресурс]. URL: https://habrahabr.ru/post/126828/ (дата обращения: 20.03.2019).

УДК 004.421

ОБЗОР ПРОГРАММ УПРАВЛЕНИЯ ЛИЧНЫМ РАСПИСАНИЕМ ПОЛЬЗОВАТЕЛЯ

А.И. Мартышкин

кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье рассматриваются некоторые существующие программы управления личным расписанием пользователя. Приводятся и анализируются известные существующие программы, в которых выделяются достоинства и недостатки. В заключение, сделан вывод о том, что все представленные в работе программы имеют избыточную функциональность, поэтому целесообразно разработать простой и функциональный органайзер.

Ключевые слова: планирование, личное расписание, органайзер, программа, оповещение, напоминание.