МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра дискретной математики и информационных технологий

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ ДОСТУПА К WEB-РЕСУРСАМ НА ОСНОВЕ ТЕХНОЛОГИИ БЛОКЧЕЙН

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 -— Информатика и вычислительная техника
факультета КНиИТ
Липкова Владислава Анатольевича

Научный руководитель
к. э. н., доцент _____ Г. Ю. Чернышова

Заведующий кафедрой
к. ф.-м. н., доцент _____ Л. Б. Тяпаев

Саратов 2024

ВВЕДЕНИЕ

В эпоху цифровых технологий, когда доступ к информации и ее безопасность становятся все более важными в различных секторах, потребность в инновационных решениях для управления web-ресурсами и их защиты возрастает. Технология блокчейн, известная своей прозрачностью, безопасностью и децентрализацией, вызывает значительный интерес у разработчиков и предпринимателей. Использование блокчейна для управления доступом к web-ресурсам открывает новые возможности для создания более безопасных и эффективных систем управления данными.

Традиционные методы аутентификации и авторизации пользователей часто имеют ряд недостатков, включая уязвимость к различным типам атак, таким как фишинг, взлом учетной записи и утечка данных. Эти методы обычно предполагают централизованное хранение данных, что увеличивает риски в случае нарушения безопасности в едином центре. Блокчейн предлагает альтернативный подход, основанный на принципах распределенного реестра и криптографии, который может радикально изменить способ управления доступом к web-ресурсам.

Внедрение технологии блокчейн для управления web-доступом предполагает создание надежной и прозрачной системы, в которой каждая транзакция или изменение прав доступа регистрируется в неизменяемом блокчейне. Это обеспечивает высокий уровень безопасности и доступности данных, а также возможность быстрой проверки истории изменений без риска подделки информации. Такие системы могут использовать смарт-контракты для автоматизации процессов проверки прав и предоставления доступа, снижения зависимости от посредников и снижения общих затрат на обслуживание системы.

Внедрение технологии блокчейн для контроля доступа к web-ресурсам требует тщательного планирования и разработки. Это включает в себя выбор подходящей блокчейн-платформы, разработку смарт-контрактов, интеграцию с существующими web-интерфейсами и обеспечение совместимости с различными типами устройств и браузеров. Важным аспектом также является эффективное масштабирование системы для работы с растущим числом пользователей и объемами данных.

Целью данной бакалаврской работы является разработка приложения для организации доступа к образовательным услугам с возможностью оплаты. Для

выполнения поставленной цели были установлены следующие задачи:

- анализ особенностей блокчейн-технологий и смарт-контрактов;
- проектирование и разработка смарт-контрактов для организации доступа к образовательным ресурсам;
- разработка бэкенд и фронтенд частей с размещением, позволяющим использовать образовательные ресурсы.

Данная бакалаврская работа включает в себя три раздела. Первый раздел посвящен обзору технологии блокчейн и смарт-контрактов. Второй раздел отведен под проектирование и разработку смарт-контрактов для оплаты услуг. Третий раздел включает в себя разработку бэкенд и фронтенд частей для образовательного ресурса.

Бакалаврская работа включает шестнадцать рисунков, две таблицы и двадцать один используемый источник.

В качестве материалов в бакалаврской работе использовались научные издания, учебники, учебные пособия, материалы научных конференций и журнальные статьи.

1 Основное содержание работы

Блокчейн можно описать как публичный реестр, в котором хранятся все совершенные транзакции в виде цепочки блоков. Эта цепочка постоянно увеличивается, когда к ней добавляются новые блоки. Технология блокчейн обладает ключевыми характеристиками, такими как децентрализация, постоянство, анонимность и возможность аудита. Блокчейн может работать в децентрализованной среде, что обеспечивается за счет интеграции нескольких основных технологий, таких как криптографический хэш, цифровая подпись и механизм распределенного консенсуса. С помощью технологии блокчейн транзакция может осуществляться децентрализованным образом [1].

В традиционных централизованных системах данные хранятся в одном месте, что делает их уязвимыми для атак и манипуляций. В блокчейне, напротив, данные дублируются и распределяются по всей сети узлов. Каждый узел имеет копию всей цепочки блоков, что обеспечивает высокую устойчивость системы к сбоям и атакам. Децентрализация устраняет необходимость в доверии к одному центральному органу, такому как банк или правительственное учреждение [2].

Начальный блок любого блокчейна называется генезис-блоком. Генезис-блок – это специальный блок, который имеет нулевой номер и жестко закодирован в каждом блокчейне. Каждый другой блок ссылается на предыдущий блок. Блокчейн растет за счет добавления новых блоков в существующую цепочку. Традиционные блокчейны состоят из транзакции, представляющие собой обмен средствами между двумя адресами. Для повышения эффективности каждая действительная транзакция записывается в блок, который может содержать несколько транзакций [3].

Клиент передает запрос на транзакцию одному из участников. Этот узел-участник осуществляет групповую рассылку запроса клиента всем остальным узлам. Как только все узлы получают копию запроса клиента, они запускают согласованный протокол. Выбор базового согласованного протокола влияет на временную сложность и потребление ресурсов. Победитель этапа утверждения предлагает следующий блок и передает его всем остальным узлам. Этот процесс передачи эквивалентен добавлению записи в глобальный распределенный реестр [4].

Для достижения согласия о состоянии реестра в децентрализованной сети используются механизмы консенсуса. PoW – это алгоритм консенсуса, основная

которого идея заключается в распределении прав на учет и вознаграждений посредством конкуренции за мощность хэширования между узлами. На основе информации из предыдущего блока различные узлы вычисляют конкретное значение хэша [5]. Первый узел, который решит эту математическую задачу, может создать следующий блок и получить вознаграждение. Конкретные этапы расчета заключаются в следующем:

В рамках консенсус PoS (Proof of Stake) цифровая валюта имеет концепцию возраста монеты. Возраст монеты – это ее стоимость, умноженная на период времени с момента ее создания. Чем дольше один узел хранит монеты, тем больше прав он может получить в сети. Владельцы монет также получает вознаграждение в соответствии с возрастом монеты. Proofhash – это хэш-значение, состоящее из весового коэффициента, неизрасходованного выходного значения и нечеткой суммы текущего времени. PoS ограничивает мощность хэширования каждого узла. Сложность майнинга обратно пропорциональна возрасту монеты. PoS поощряет владельцев монет увеличивать время хранения. С появлением концепции возраста монеты блокчейн больше не будет полностью полагаться на доказательство работоспособности. Это эффективно решает проблему растраты ресурсов в PoW. Безопасность блокчейна, использующего PoS, повышается с увеличением стоимости в блокчейн. Злоумышленникам необходимо накопить большое количество монет и удерживать их достаточно долго, чтобы атаковать блокчейн. Это также значительно увеличивает сложность атаки [6].

Далее в работе будет использоваться блокчейн Ethereum, который имеет схожую архитектуру с блокчейном Bitcoin. Основное различие заключается в том, что блоки Ethereum содержат копию как списка транзакций, так и самого последнего состояния. Помимо этого, в блоке также хранятся два других значения – номер блока и сложность. Ethereum объединенил и улучшил концепцию сценариев, альткоинов и сетевых технологий протоколов и предоставил разработчикам возможности создавать произвольные приложения на основе консенсуса, которые имеют масштабируемость, стандартизацию, полноту функций, простоту разработки и совместимость, предлагаемые этими разными парадигмами одновременно. Ethereum имеет встроенный язык программирования, полный по Тьюрингу, позволяющий любому писать смарт-контракты и децентрализованные приложения, где есть возможность создавать свои собственные произвольные правила владения, форматы транзакций и функции переходы со-

стояний [7].

Код в контрактах Ethereum написан на низкоуровневом языке байт-кода, основанном на стеке, который называется код виртуальной машины Ethereum (EVM). Код состоит из последовательности байтов, где каждый байт представляет операцию. В общем, выполнение кода представляет собой бесконечный цикл, состоящий из многократного выполнения операции с текущим программным счетчиком (который начинается с нуля) и последующего увеличения программного счетчика на единицу до тех пор, пока не будет достигнут конец кода или не будет обнаружена ошибка, команда остановки или возврата. Транзакция в EVM – это криптографически подписанный пакет данных, хранящий сообщение, в котором EVM сообщает о необходимости передачи, создания нового контракта, запуска существующего контракта или выполнения некоторых вычислений. Адреса контрактов могут быть получателями транзакций, как и пользователи с внешними учетными записями [8].

Смарт-контракт – это набор обещаний, указанные в цифровой форме, включающие протоколы, в рамках которых стороны выполняют эти обещания [9]. Основная идея смарт-контрактов заключается в том, что многие виды договорных положений (таких как залоги, привязки, разграничение прав собственности) могут быть встроены в аппаратное и программное обеспечение [10]. Токен – шаблон используемый для распространения некоторых взаимозаменяемых товаров (представленных токенами) среди пользователей. Токены могут представлять собой широкий спектр товаров, таких как, например, монеты, акции, результаты или билеты, или все остальное, что можно передавать и подсчитывать. Последствия владения токеном зависят от протокола и варианта использования, для которого был выпущен токен. Токены могут использоваться для подтверждения владения физическими объектами или цифровыми объектами. Токены также используются для регулирования авторизации и идентификации пользователей.

Для создания смарт-контрактов была выбрана совместимая с Ethereum Virtual Machine сеть для разработки и реализации децентрализованных приложений. Причинами выбора являются высокая степень безопасности, хорошо развитая инфраструктура и широкая поддержка разработчиков. Для написания смарт-контрактов использовался язык программирования Solidity. В качестве фреймворка был выбран Foundry. Для компиляции и тестирования из команд-

ной строки была утилита forge, которая поставляется вместе с foundry.

Входной точкой для пользователей является смарт-контракт ServiceMarket, который занимается менеджментом услуг и позволяет пользователям покупать услуги. Данный контрат должен иметь возможность обновления, так как с течением времени могут меняться требования для внутренней логики контракта. Каждая отдельная услуга должна быть представленна отдельным смарт-контрактом, логика которых может отличаться, но наследоваться от общего интерфейса IService. Данные контракты должны представлять собой представлять фабрику контрактов, которые представляют собой невзаимозаменяемый токены, предоставляющие доступ к услугам. Данные контракты должны наследоваться от интерфейса ITicket и имплементировать методы из стандарта IERC721.

Для обеспечения авторизации и управления доступом к услугам использовался стандарт ERC721, где каждый тикет — это NFT, предоставляющий уникальные права на доступ к конкретным услугам. Разработка началась с написания контракта ServiceMarket, который агрегирует услуги и управляет их взаимодействиями. Далее были разработаны контракты для каждой услуги, соответствующие интерфейсу IService.

При разработке смарт-контрактов для сети Ethereum стоит учитывать существенные отличия от разработки любого другого программного обеспечения, связанных с особенностями и ограничениями виртуальной машины Ethereum. Одно из важных ограничений – это лимит использования локальных переменных в функциях. Это ограничений связано с архитектурой EVM, которая предоставляет ограниченный стек для выполнения операций. Каждая переменная занимает место в стеке, а для добавления получения конкретного элемента используется инструкция PUSHα и DUPα, где $\alpha \in [1, 16]$.

Для обеспечения безопасности и надежности была использована библиотека OpenZeppelin. В данной библиотеке уже реализованы популярные шаблоны, которые прошли аудит. ServiceMarket является точкой входа для всех взаимодействий пользователей. Данный контракт должен позволять пользователям покупать выбранную услугу, а администраторам давать возможность добавление или удаление услуг. При маштабировании логика методов может быть изменена, но данные должны сохранятся. Для реализации данного требования ServiceMarket унаследован от контракта UUPSUpgradeable. Также доступ к ад-

министративным функциям контракта не должен быть привязан к одному конкретному адресу, поэтому все права буду переданы контракту AccessManager, который будет выдавать права для выполнения конкретных административных функций.

Контракты Service, могут иметь разную логику, но должны быть реализованы все методы интерфейса IService. Данные контракты должны быть фабрикой контрактов. При получении ERC20 токенов сервис чеканит собственную реализацию ERC721 токена и сохранят адрес, кому принадлежит этот токен. Также у контракта должна быть возможность переводить все полученные средства на адрес владельца. ITicket унаследован от стандарта ERC721, который обеспечивает соответствие функциональным возможностям ERC721, таким как отслеживание владения и передача токенов. Это наследование позволяет ITicket использовать все свойства и методы токена ERC721, что делает его подходящим для представления уникальных активов или прав доступа. ITicket предназначен для хранения дополнительных данных, связанных с каждым токеном. Это может включать такие сведения, как временная метка покупки, дата истечения срока действия услуги и конкретные атрибуты, относящиеся к предоставляемой услуге. Например, если услуга предполагает доступ к цифровому контенту, токен может хранить хэш контента или ключ дешифрования. Контракт Service, действующий как фабрика, обязывает ITicket чеканить новые токены всякий раз, когда услуга приобретается за токены ERC20. В процессе создания ITicket регистрируется не только владельц, но и дополнительные параметры, переданные во время транзакции. Это гарантирует, что каждый токен уникален и содержит всю необходимую информацию для доступа к сервису. Фрагменты программной реализации смарт-контрактов представлены в приложении А.

После тестирования смарт-контракты были развернуты в сети BSC. Этот процесс включал окончательную проверку и оптимизацию кода для минимизации затрат на газ и обеспечения высокой производительности блокчейна. Развертывание и мониторинг в режиме реального времени были облегчены благодаря использованию Tenderly, что помогло быстро выявить и устранить любые проблемы после развертывания.

Этапы разработки и внедрения смарт-контрактов демонстрируют сложность и многоуровневость процесса создания безопасных и эффективных блокчейн-решений для образовательной платформы.

Данный сервис позволяет авторизовываться с помощью web3-кошелька. Получать доступ к уже существующим курсам или создавать новые. Для получения доступа необходимо оплатить курс, нажав на кнопку оплатить, после чего сформируется транзакция на оплату, которую необходимо будет только подпсать с помощью кошелька. После данных действий пользователь получит доступ к курсу. Также у автора курса есть возможность забрать всю награду полученную от других пользователей, купивших курс.

Бэкенд часть проекта образовательных ресурсов была разработана для интеграции образовательных курсов с технологией блокчейн, в частности, с использованием кошельков Web3 для аутентификации пользователей и смарт-контрактов для оплаты курсов.

Система разделена на две основные области: управление пользователями и управление курсами, каждая из которых инкапсулирована в специальные службы и хранилища.

Фронтенд образовательного ресурса разработан с использованием Next.js, современного фреймворка для React, который предоставляет серверный рендеринг и генерацию статических сайтов. Это обеспечивает высокую производительность и оптимизацию SEO. Для управления взаимодействием с блокчейн используется библиотека wagmi в сочетании с RainbowKit, что позволяет обеспечить поддержку различных кошельков и упрощает процесс авторизации и выполнения транзакций в Ethereum сети.

Каждый компонент системы, включая бэкенд и фронтенд, упакован в отдельные Docker-контейнеры. Контейнеры настраиваются для выполнения в изолированной среде с точно определенными зависимостями, что устраняет проблемы совместимости и упрощает процесс развертывания.

ЗАКЛЮЧЕНИЕ

Использование блокчейн-технологий в web-технологиях открывает новые перспективы для множества отраслей, включая образование, финансы, юриспруденцию и медиа. Эти технологии обеспечивают улучшенную инфраструктуру для создания открытых систем, в которых каждый пользователь может верифицировать подлинность и правомерность операций без необходимости дополнительных подтверждений.

Интеграция блокчейн в системы управления доступом к web-ресурсам представляет собой значительный шаг вперед в создании более безопасных и устойчивых цифровых сервисов. Она обеспечивает необходимую инфраструктуру для построения децентрализованных приложений, где безопасность и прозрачность становятся ключевыми факторами успеха.

В бакалаврской работе был проведен анализ особенностей технологии блокчейен и смарт-контрактов, спроектированы и разработаны смарт-контракты для организации доступа к образовательным курсам, разработана бэкенд и фронтенд часть с размещением в Google Cloud.

В процессе разработки были созданы и протестированы смарт-контракты для оплаты образовательных услуг, что позволило интегрировать криптовалютные платежи непосредственно в структуру образовательного ресурса. Разработанные контракты обеспечивают не только обработку транзакций, но и управление доступом к курсам, что значительно расширяет возможности для образовательных платформ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Zheng Z. et al. Blockchain challenges and opportunities: A survey //International journal of web and grid services. – 2018. – Т. 14. – №. 4. – С. 352-375.

2 Victor F., Lüders B. K. Measuring ethereum-based erc20 token networks //Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23. – Springer International Publishing, 2019. – С. 113-129.

3 Nakamoto S. et al. Bitcoin //A peer-to-peer electronic cash system. – 2008. – Т. 21260.

4 Katz J., Lindell Y. Introduction to modern cryptography: principles and protocols. – Chapman and hall/CRC, 2007.

5 Lee Kuo Chuen D. Handbook of digital currency. – Elsevier, 2015.

6 King S., Nadal S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake //self-published paper, August. – 2012. – Т. 19. – №. 1.

7 Buterin V. A next-generation smart contract and decentralized application platform //white paper. – 2014. – Т. 3. – №. 37. – С. 2-1.

8 Dannen C. Introducing Ethereum and solidity. – Berkeley : Apress, 2017. – Т. 1. – С. 159-160.

9 Szabo N. Smart contracts: building blocks for digital markets //EXTROPY: The Journal of Transhumanist Thought,(16). – 1996. – Т. 18. – №. 2. – С. 28-39.

10 Bauer D. P. ERC721 nonfungible tokens //Getting Started with Ethereum: A Step-by-Step Guide to Becoming a Blockchain Developer. – Berkeley, CA : Apress, 2022. – С. 55-74.