Санкт–Петербургский государственный университет

Цепелева Рита Вячеславовна

Выпускная квалификационная работа

Имплементация протокола по кросс блокчейн взаимодействию

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2015 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Математическое и программное обеспечение вычислительных машин»

Научный руководитель:
доцент, кафедра компьютерного моделирования
и многопроцессорных систем, к.ф.-м.н.
Корхов Владимир Владиславович

Рецензент:
директор по развитию, ООО "Манзони
к.т.н. Тихомиров Владимир Александрович

Санкт-Петербург
2021 г.

Содержание

Введение

На сегодняшний день блокчейн (распределенный реестр, данные в котором хранятся в блоках, создающих последовательную непрерывную цепочку)[1] является одной из самых популярных и перспективных технологий в IT-индустрии. Блокчейн имеет огромное количество применений, одним из которых являются децентрализованные финансы (DeFi)[2]. Основной задачей децентрализованных финансов является создание альтернативы банковскому сервису. А именно замена традиционных банковских операций протоколами с открытым исходным кодом. В настоящий момент концепция DeFi активно развивается и популяризируется. Убедиться в этом можно, посмотрев на график зависимости совокупной суммы средств, заблокированных в протоколе DeFi (график показателя total value locked, TVL)[3](Рис.1).

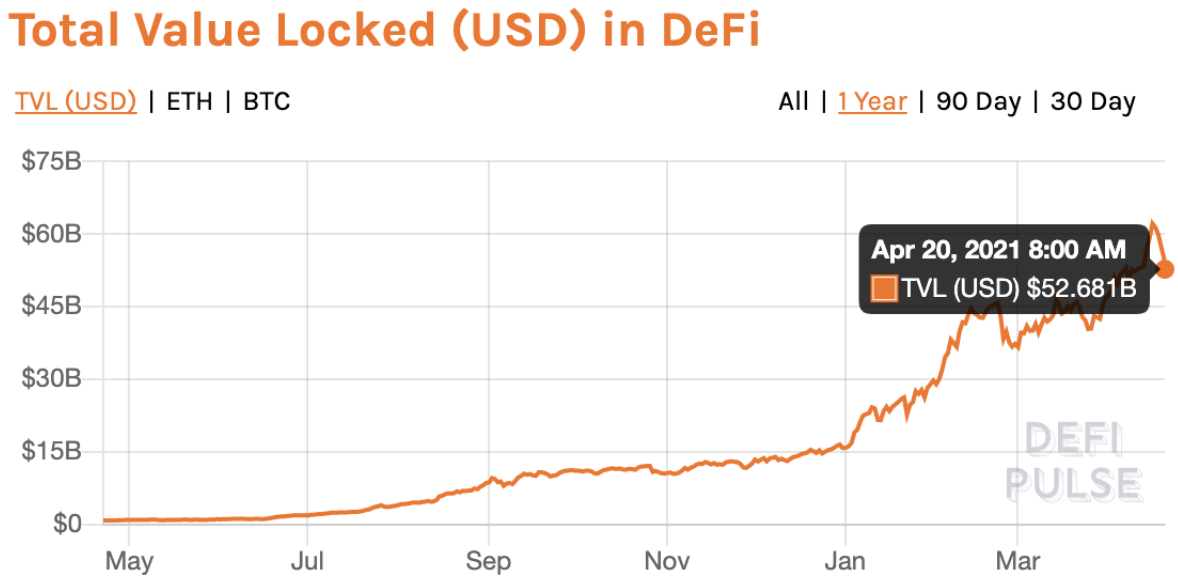

Рис. 1: Рост средств, заблокированных в протоколе DeFi.

История децентрализованных финансов берет свое начало в октябре 2017 года. Концепция DeFi практически сразу стала трендом и оживила индустрию криптовалют идеей о том, что предприниматели смогут обходить централизованный контроль банков, компаний и правительств. Как мы можем видеть, на данный момент в системе уже заключено более 52-ух

миллиардов долларов. Индустрия привлекает огромное количество финансовых средств и пользователей, несмотря на внутренние проблемы, которые, откровенно говоря, являются серьезными, но вполне разрешимыми. Решение и обзор одной из таких существенных проблем представлено в данной выпускной квалификационной работе.

Постановка задачи

Проблема концепции децентрализованных финансов заключается в следующем. Изначально индустрия DeFi задумывалась как единая экосистема, продукты которой будут легко совместимы между собой. Но дело в том, что каждая блокчейн-сеть создается самостоятельной, независимой и автономно работающей. В этом и заключается сложность: блокчейны сами по себе не могут взаимодействовать друг с другом и передавать друг другу информацию без посторонней помощи. Следовательно, ни о какой единой экосистеме речи идти не может. Именно поэтому вопрос объединения блокчейн сетей актуален вот уже несколько лет. Хочется также отметить, что данная задача распространяется не только на индустрию децентрализованных финансов, но и на практически все сферы применения технологии. Ведь сегодня блокчейн-разработчики при разработке и дизайне своих децентрализованных приложений вынуждены выбирать лишь одну блокчейн-платформу для своего будущего продукта и жертвовать такими серьезными и важными показателями как производительность, скорость, безопасность, масштабируемость и т.д., вместо того, чтобы объединять в своем проекте несколько блокчейн-сетей и использовать каждую в соответствии с ее преимуществами.

Данную проблему логичнее всего было бы решить, написав смарт-контракты[4] – программы на высокоуровневых, Тьюринг-полных языках программирования. Основным отличием данного программного кода от иного является невозможность редактирования или изменения после публикации его в блокчейне. Именно этот факт вызывает определенную сложность в написании смарт-контрактов и требует хорошего знания всех нюансов технологии блокчейн, сетей, с которыми ведется работа, а также обладания навыком программирования на таких языках, как, например, Solidity[5].

Таким образом, нашей задачей становится нахождение и реализация такого решения, которое бы позволило быстро, дешево и безопасно передавать токены (цифровые активы) между блокчейн сетями. Также необходимо предоставить клиентам такой интерфейс, благодаря которому каждый

технически неподготовленный пользователь смог бы воспользоваться нашим решением.

Для достижения этой цели необходимо выполнить следующие подзадачи:

- подробно изучить технологию блокчейн и индустрию децентрализованных финансов;
- проанализировать существующие решения, выявить их преимущества и недостатки;
- разобраться в структуре и принципах работы существующего сервиса компании, предложившей решить выявленные проблемы в качестве выпускной квалификационной работы;
- основываясь на полученных знаниях предложить идею решения, выдвинуть технические требования, подобрать архитектуру проекта и инструменты для его реализации;
- реализовать минимально жизнеспособный продукт, который будет удовлетворять поставленным требованиям;
- проанализировать полученное решение, сравнить с описанными существующими продуктами и подходами.

Обзор литературы

Статьи [1], [13] и [18] содержат подробное описание различных аспектов и основную идею технологии блокчейн.

Публикации [2], [10], [11] и [21] описывают индустрию децентрализованных финансов и различные ее составляющие (децентрализованные биржи, теории автоматического маркет-мейкинга, стейблкоины и т.д.). Это помогает понять общую идею экосистемы, выявить проблемы и разработать собственный план по их решению.

Документ [8] и ресурс [12] содержат официальную документацию экосистемы Polkadot, а статьи [15], [16] и [17] помогают углубиться в работу технологии и изучить систему более детально.

Статья [22] дает возможность разобраться в теме блокчейн-мостов, в сути их разработки и в том, как сделать их безопасными для пользователей.

Ресурсы [19] и [23] содержат ссылки на сервисы, предоставляющие уже готовые решения заявленной проблемы.

Основные документации сетей Ethereum[4], Binance[6], Tron[7], NEO[9] и других содержат в себе технические спецификации сетей и помогают сделать выбор конкретных блокчейн-платформ для осуществления кроссчейн обмена.

И наконец ресурсы [5], [25], [26], [28], [29] и [31] содержат ссылки на инструменты, ознакомление с которыми просто необходимо для решения поставленной задачи.

Помимо вышеперечисленных статей, документаций и тд. существуют еще и другие ресурсы, помогающие расширить свои знания в области технологии блокчейн и существующих в ней разработок.

Глава 1. Децентрализованные финансы

1.1 Определение и суть концепции

Децентрализованные финансы (Decentralized finances, DeFi) – это независимая экосистема, предоставляющая пользователям финансовые сервисы, работающие на публичных блокчейнах. При этом основной целью экосистемы, как уже было сказано ранее, является построение альтернативы традиционной финансовой системы, но без посредников и государственного контроля.

Примерами таких децентрализованных финансовых сервисов являются кредитование, депозиты, страхование, децентрализованные биржи и так далее. Преимуществом использования таких децентрализованных аналогов обычных банковских операций заключается в сохранении пользователями полного контроля над своими активами и взаимодействие с экосистемой через одноранговые (peer-to-peer, P2P), децентрализованные (DApps) приложения. То есть участники самостоятельно справляются с совершением сделки, что экономит их время, деньги, а также позволяет сохранить конфиденциальность.

Основной сетью, где располагается большинство DeFi-приложений считается Ethereum, но помимо него блокчейн-разработчики для своих приложений выбирают и такие сети, как Binance Smart Chain[6], Tron[7], Polkadot[8], NEO[9] и другие.

Сервисы децентрализованных финансов работают на основе смарт-контрактов с открытым исходным кодом, которые значительно уменьшают влияние человеческого фактора в предоставлении данного рода услуг, однако не могут полностью гарантировать безопасность. Ведь всем известен факт того, что незамеченная разработчиками ошибка или уязвимость в коде контракта может стоить им и их пользователям огромных денег. Например, в марте 2021 года сервис Furucombo, который позволяет создавать и объединять транзакции в различных протоколах DeFi, подвергся атаке. Злоумышленники вывели ETH и различные токены на сумму более 14 миллионов долларов. Но для предотвращения таких ситауций существуют

инструменты по поиску уязвимостей и аудит смарт-контрактов. Использование данных инструментов и обращение к аудиторам помогают избежать проблем и существенно снизить риски возникновения непредвиденных ситуаций.

1.2 Варианты использования сервисов DeFi

Разработчики уже предложили достаточно много вариантов использования децентрализованного финансирования, например:

- **Кредитование.** Один из самых популярных видов приложений DeFi. Основными преимуществами открытых децентрализованных займов перед традиционной банковской услугой кредитования являются: быстрый расчет платежей, обеспечение цифровыми активами, отсутсвие кредитных проверок, а также полное доверие к технологии ввиду расположения сервиса на публичном, открытом блокчейне. Благодаря данным характеристикам, децентрализованное кредитование способно предоставить пользователям более дешевый, быстрый, а также доступный для большего числа людей сервис;

- **Денежно-банковские услуги.** Это может быть очень обширный спектр услуг, которые предоставляют возможность пользователям, например, взять ипотеку или оформить страховку с использованием протоколов DeFi. Они могут значительно уменьшить количество посредников, что позволит снизить как финансовые, так и временные затраты пользователей. Кроме того в данной сфере очень популярны приложения по выпуску стейблкоинов[10] (криптовалюта, созданная для имитации курса фиатных валют), которые потенциально могли бы стать хорошей альтернативой обыкновенным, привычным нам электронным деньгам. Ведь их преимущество заключается, опять же, в том, что они не эмитируются государством, а значит полностью подконтрольны своим владельцам.

- **Трейдинг.** Одним из важнейших типов DeFi-приложений в данной области являются децентрализованные биржи (DEX)[11], которые поз-

воляют пользователям торговать своими цифровыми активами без всяких посредников. Сделки на DEX осуществляются через смарт-контракты между пользователями напрямую, что снижает потребность торговой платформы в большом объеме технического обслуживания. Соответственно и комиссии за оказаные услуги децентрализованные биржи устанавливают гораздо меньшие, чем их традиционные аналоги.

Помимо вышеперечисленных популярных вариантов использования децентрализованных финансовых инструментов, существуют и иные проекты, способные создавать деривативы, синтетические активы, рынки децентрализованного прогнозирования и многое другое.

Глава 2. Обзор существующих решений и подходов

Как уже было сказано, проблема объединения и взаимодействия блокчейнов является актуальной на сегодняшний день. Именно поэтому на данный момент уже существуют готовые или относительно готовые проекты по ее решению. Рассмотрим их подробнее.

2.1 Экосистема Polkadot

2.1.1 Структура проекта и внутреннее устройство

Одним из решений проблемы передачи токенов (и другой информации) между блокчейн сетями является использование технологии Polkadot[12]. Polkadot - это сеть, которая соединяет блокчейны. Она создает пространство, где могут быстро и безопасно обмениваться и обрабатываться данные из разных блокчейнов. Polkadot призвана решить ключевые проблемы, препятствующие технологии блокчейн стать полномасштабным практическим приложением, а именно проблемы с масштабируемостью и изоляцией. Ведь на сегодняшний день не все популярные блокчейн-сети способны обрабатывать большое количество транзакций за приемлемое количество времени. Кроме того, все блокчейны, как уже было сказано ранее, не могут передавать друг другу информацию, что является сильным ограничением как для пользователей, так и для разработчиков.

Структура Polkadot изображена на рис. 2. Она включает следующие компоненты:

- **Relay Chain (главная цепь или связующая цепь).** Это основная цепь Polkadot, соединяющая все блокчейны в сети;
- **Parachain (парачейн, параллельная цепь).** Для осуществления транзакций и переноса их в исходный блокчейн используются параллельные блокчейны. Парачейны строят коллаторы: их цель - сбор транзакции пользователей и подтверждение блоков через алгоритм доказательства валидности. Коллаторов награждают за работу. В зависимости от конкретного парачейна определяется размер награды.

Деятельность коллаторов схожа с работой майнеров в блокчейнах (с использованием алгоритмов Proof-of-Stake, Proof-of-Work)[13];

- **Bridge (мост).** Мосты предназначены для соединения блокчейнов, не использующих протоколы управления Polkadot (например, блокчейны Биткоина или Эфириума).

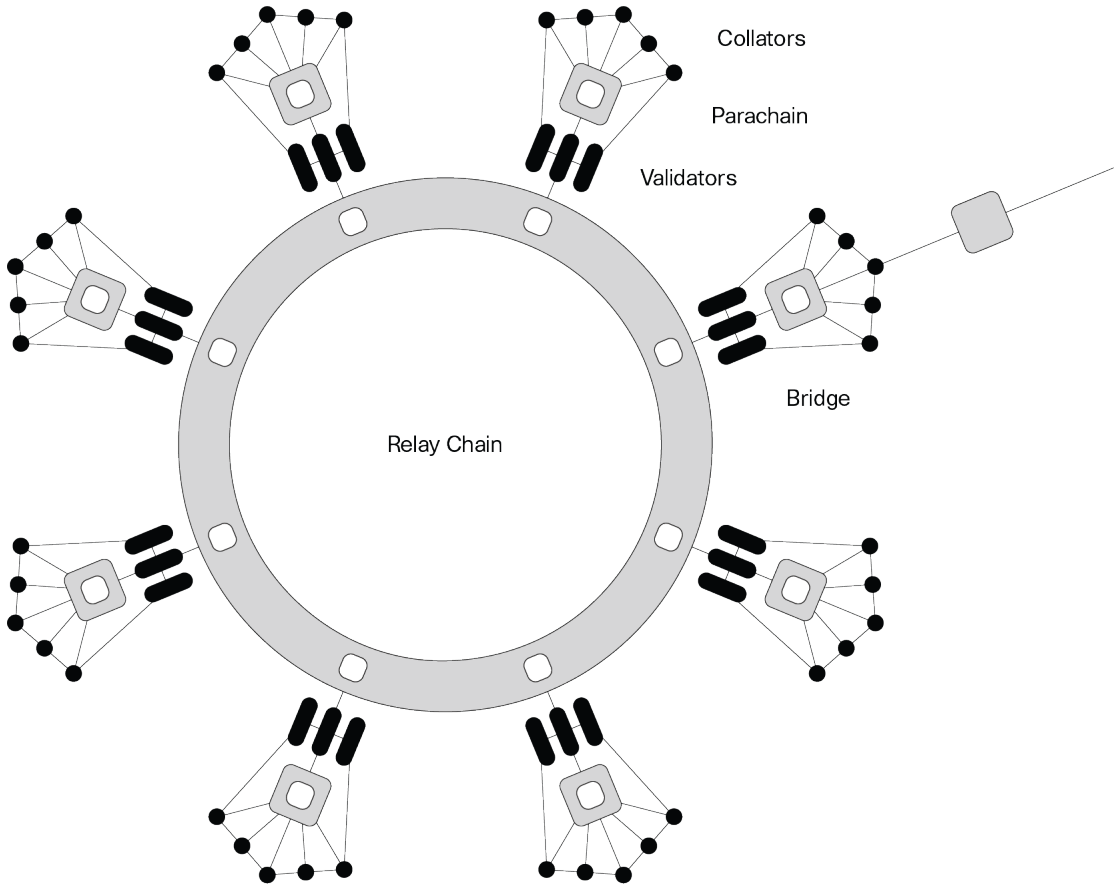

Рис. 2: Структура экосистемы Polkadot.

Главная цепь выполняет во всей системе функции контроля за соблюдением консенсуса (соглашения о едином состоянии сети), отвечает за доставку сообщений между цепями-участниками, а также способствует финализации транзакций. Сама по себе главная цепь - это блокчейн с множеством валидаторов, которые каждый раз выбираются случайным образом для проведения валидации и добавления новых блоков в парачейны. Чтобы

выполнить свои обязательства, валидаторы для каждой транзакции вносят депозит, который возвращается в случае соответствия транзакции правилам консенсуса. В противном случае депозит исчезает. Валидаторы за свою работу получают определенное вознаграждение.

Polkadot в своей реализации не используют классические алгоритмы достижения консенсуса, такие как Proof-of-Work или Proof-of-Stake. Вместо этого Polkadot использует гибридный консенсус, достижение которого осуществляется посредством двух механизмов: производства блока и финализации. Благодаря тому, что команда Polkadot в своем проекте распараллелили эти два процесса, они смогли добиться более быстрой генерации блоков. Таким образом в Polkadot решается проблема ограничений масштабируемости в протоколах с мгновенной финализацией, таких как Tendermint[14].

За генерацию блоков в сети Polkadot отвечает протокол BABE (Blind Assignment for Blockchain Extension)[15]. Данный алгоритм имеет сложность $O(n)$ и работает на основе временных интервалов, предоставляя право на производство блока в главной цепи слот-лидеру. Слот-лидером в Polkadot называется валидатор, случайно выбранный алгоритмом BABE в процессе проведения вычислений с использованием верифицируемой случайной функции. Таким образом BABE создает блоки в главной цепи, состоящие из заголовков всех доступных в сети блоков.

За финализацию же в Polkadot отвечает протокол GRANDPA (GHOST-based Recursive Ancestor Deriving Prefix Agreement)[16]. В Polkadot, как известно, валидаторы голосуют своими депозитами (стейками) за самый высокий блок, который они считают валидным. И каждый блок, набравший более двух третей таких голосов, финализируется. Алгоритм GRANDPA имеет сложность $O(n^2)$. Это значит, что при увеличении количества узлов в два раза, вам придется отправить в четыре раза больше сообщений. Так вот Polkadot в своей реализации позволяет пулу валидаторов отправлять свои голоса и финализировать сразу несколько блоков за раз в главной цепи, в отличие от других консенсусных систем, где производство блоков является частью процесса финализации и отправка сообщений происходит для каждого блока отдельно. Это существенно замедляет весь процесс ге-

нерации блоков.

В системе Polkadot валидаторов утверждают номинаторы (пассивные участники PoS-экосистемы Polkadot), которые вносят за них залог. Если валидатора уличат в нарушении правил консенсуса, залог аннулируется. Если валидатор все таки оправдает доверие, то номинатор получит депозит обратно и заработает вознаграждение. Такой алгоритм достижения консенсуса называется Nominated Proof-of-Stake (NPoS)[17]. Он отличается от традиционных алгоритмов Delegated Proof-of-Stake (DPoS)[18], ведь здесь номинаторы не участвуют в поддержании работоспособности и безопасности сети. В их обязанности входит лишь выбор надежного валидатора и стейк токенов.

А за безопасностью и поведением валидаторов, в свою очередь, следят так называемые стороженые ноды, которые, при выявлении противоречивого поведения валидатора создают определенные доказательства, содержащие информацию о нарушении. Сторожевые ноды получают часть конфискованных стейков валидаторов. Такая система позволяет GRANDPA обеспечивать асинхронную подконтрольную безопасность: финализация двух любых конфликтующих блоков приведет к тому, что ответственные валидаторы теряют свои депозиты. Протокол GRANDPA используется во всех парачейнах сети Polkadot.

2.1.2 Проекты на Polkadot

На основе данной технологии основано много различных проектов. Некоторые из них по своей сути и цели частично или полностью совпадают с тем, что хотим сделать мы. Например, Polkaswap[19]. В первую очередь Polkaswap - это децентрализованная биржа для экосистемы Polkadot, на данный момент работающая в тестовом режиме на блокчейне SORA[20]. Она предоставляет структуру, которая соединяет множество блокчейнов с помощью мостов и позволяет участникам Polkadot и пользователям других блокчейнов подключаться и вести быструю и эффективную торговлю своими активами. Такие показатели достигаются благодаря интеграции с технологией Polkadot, а также благодаря использованию протокола авто-

матического маркет-мейкинга(AMM)[21] сразу для нескольких пулов ликвидности.

Понятие "пула ликвидности"в DeFi, а точнее сфере трейдинга и децентрализованных бирж(DEX), является ключевым и из себя представляет аналог привычного нам банка, хранящего огромное количество денег, которым имеет право распоряжаться так, чтобы получить прибыль. Пулы ликвидности, как и банки, хранят в себе средства в нескольких валютах (традиционно число валют равно двум). И то соотношение, которое они формируют, определяет курс токена. Но пулы ликвидности нужны не только для того, чтобы регламентировать курс. Их основной задачей является предоставление средств для обеспечения обмена валют. Если, например, пользователь захочет обменять ETH на токен DAI, он, используя интерфейс децентрализованной биржи, заберет токен DAI из пула, а свой ETH внесет туда. Но ведь если произвести подобную операцию, очевидно, что соотношение ETH к токену DAI изменится (ETH в пуле ликвидности стало больше, а DAI - меньше). Ведь до обмена в хранилище лежало, предположим, 100 ETH и 250000 токенов DAI. Это значит, что 1 ETH можно было обменять на 2500 DAI. Пусть пользователь так и сделал. Это привело к тому, что в пуле теперь оказалось 101 ETH и 247500 DAI. Курс изменился. Для того, чтобы держать эту ситуацию под контролем и предоставлять ликвидность децентрализованные биржи полагаются на автоматических маркет-мейкеров(AMM). AMM - это смарт-контракты со встроенным специально разработанным алгоритмом, который по математическим формулам ведет расчеты, устанавливает цены, создает пулы ликвидности.

Ранее, вместо децентрализованных бирж использовались так называемые книги заказов (order book). Они выглядят как таблицы, их строки содержат желания пользователей обменять один токен на другой по определенному курсу. Такая система предложений по обмену цифровыми активами уже давно изжила себя. Ведь при сравнении с DEX можно увидеть недостатки этой модели, основным из которых является время ожидания. Пользователь, разместив свой трейд в книге заказов мог ждать несколько дней, пока найдется человек, согласный на условия сделки. И это большая удача, если он вообще найдется. Естественно, бывали случаи, когда срок

заказа истекал и пользователь терял свое время на ожидание впустую. В этом плане децентрализованные биржи определенно выигрывают. Ведь их цель – предоставить ликвидность и обеспечить пользователям возможность обменивать свои активы быстро и эффективно.

Однако, не все зависит только от концепции. На сегодняшний день большинство децентрализованных бирж расположены в сети Ethereum, которая, как всем известно, имеет большие проблемы с масштабируемостью. Это отражается на пользователях, которые вынуждены платить высокие комиссии за газ, а также ждать определенное количество времени, пока пройдет транзакция (сеть Ethereum имеет относительно низкую пропускную способность 15-45 транзакций в секунду). Это является серьезной проблемой в биржевой торговле, ведь здесь действительно важна каждая секунда.

Polkaswap, в свою очередь, предлагает решение данных проблем, создавая децентрализованную биржу на основе технологии Polkadot, которая позволит не только снизить комиссии за операции и время ожидания, но и сделает возможным перемещение цифровых активов между различными сетями (это объединяет идейно наши проекты).

Единственным выявленным недостатком не только проекта Polkaswap, но и всех проектов на основе Polkadot, является безопасность. При более глубоком изучении вопроса о том, как именно происходит межсетевой обмен токенов, выяснилось, что Polkadot в своей системе использует обыкновенные адреса (кошельки) для хранения токенов, а не смарт-контракты. Здесь логика обмена осуществляется следующая: если пользователь хочет обменять токены, он, используя интерфейс Polkadot, переводит свои активы на обычный адрес-хранилище в сети отправителя. Далее в целевой сети аналогично должна произойти транзакция из хранилища на адрес пользователя, переводящая эквивалентное количество токенов. Но так как здесь используется обыкновенный адрес, а не смарт-контракт с открытым исходным кодом, пользователь не может быть уверен, что токены, которые он отправил, не осядут в хранилище, и, что надлежащие количество токенов в другой сети действительно будет ему начислено.

2.2 Блокчейн-мосты

Еще одним решением для кроссчейн передачи токенов является блокчейн-мост[22]. Блокчейн-мост – это звено, связывающее две сети блокчейна. Использовать мосты можно не только для передачи цифровых активов, но и для передачи любой другой информации. Это значит, что блокчейн-мосты делают возможным использование разработчиками сразу нескольких блокчейн-платформ при дизайне своих децентрализованных приложений, что существенно улучшит масштабируемость и иные технические характеристики проектов.

Передача токенов между сетями через блокчейн-мосты регулируется протоколом mint-and-burn. То есть для того, чтобы осуществить передачу токена из одной сети в другую, токен необходимо сжечь в исходной сети и выпустить заново в целевой сети. Такой механизм позволяет держать под контролем количество выпущенных токенов и сохранять определенную пропорцию.

Примером такого моста является Panama Bridge[23]. Panama Bridge – это блокчейн-мост между сетями Binance Smart Chain и Ethereum. Он позволяет переносить цифровые активы пользователей между сетями. Проект изначально был создан для обеспечения ликвидности децентрализованных приложений в сети Binance Smart Chain (BSC) и Binance (в сети Binance расположена одна их крупнейших децентрализованных бирж в мире). На данный момент проект развивается, подключая другие сети.

Panama Bridge также предоставляет API, что позволяет разработчикам использовать мост у себя в проекте, собирая POST/GET запросы и передавая их мосту. Это было бы быстрым решением нашей проблемы, однако надо учесть тот факт, что сервис Panama Bridge берет комиссию с пользователей за предоставление услуги, что естественно. Наша компания также планирует осуществлять сбор комиссии. Соответственно, ни один пользователь не станет платить двойную комиссию, а будет пользоваться оригинальным сервисом.

Помимо этого необходимо отметить то, что использование протокола mint-and-burn влечет за собой определенные проблемы для владельцев

токенов. А именно: очевидным является тот факт, что постоянная чеканка/сжигание токенов может привести к обесцениванию цифрового актива, так как эти процессы сложно ограничить в условиях межсетевого взаимодействия. Хорошей практикой в создании токенов считается установление верхней границы, больше которой не может становится параметр total supply, отвечающий за учет всех выпущенных токенов. Взаимодействуя с такими токенами, контракт мост, располагаясь в одной сети, не сможет проверить данный лимит в какой-либо другой сети и тогда легко может произойти следующая ситуация: пользователь перечислит токены на контракт, который их уничтожит, а другой контракт не сможет выпустить ему в целевой сети необходимое количество токенов из-за верхней границы. В итоге пользователь останется ни с чем.

Глава 3. Формирование концепции нового решения

Компания, которая предложила решить ранее поставленную задачу в качестве выпускной квалификационной работы называется MyWish[24]. Основной деятельностью данной организации является предоставление услуг по автоматизированному развертыванию смарт-контрактов в различных блокчейн-сетях. На данный момент MyWish работает с такими сетями как Ethereum, Tron, NEO, Waves, Binance, Matic, EOS и предлагает готовые шаблоны для 10 видов смарт-контрактов. Помимо основного направления MyWish развивает и другие сервисы на своей платформе. Например, Airdrop – сервис, который предоставляет возможность распространить токены стандарта ERC20[25] сразу на тысячи адресов за одну транзакцию. Проект, решающий задачу, поставленную в данной ВКР, планируется запустить как один из таких сервисов.

3.1 Идея решения Wish Swap

Формализуем задачу: имеется токен WISH стандарта BEP2. Токен расположен в сети Binance, что сильно ограничивает его применение. Необходимо создать решение, позволяющее пересылать токен в другие сети и обратно. Изучив все вышеизложенные доступные методы и подходы к решению задачи о межсетевом переводе средств с учетом всех достоинств и недостатков была сформирована идея собственного решения под названием Wish Swap, которая заключается в следующем. Предлагается написание контрактов аналогов токена WISH для сетей, с которыми планируется кроссчейн взаимодействие, а также написание блокчейн-мостов, которые будут представлять из себя смарт-контракты, взаимодействующие с контрактами аналогов токена WISH. Также предлагается установить взаимодействие между контрактами в различных сетях посредством бэкенда, написанного на языке Python.

3.2 Выдвижение технических требований

Учитывая изученные существующие методы решения по кроссчейн передаче токенов, а также опираясь на предложенную идею, к сервису выдвигаются следующие технические требования, описывающие необходимый функционал сервиса:

- смарт-контракт мост будет хранить на своем счету токены WISH;
- смарт-контракт мост должен распределять хранящиеся на нем средства в процессе кроссейн-обмена с помощью бэкенда;
- смарт-контракт токена должен быть написан по стандарту ERC20 с использованием библиотеки OpenZeppelin[26] и иметь все стандартные функции и события (стандарт ERC20 выбран для полного соответствия аналогов токена начальному токену, имеющему стандарт BEP2);
- смарт-контракт мост должен иметь доступ к токенам для операций с ними;
- все смарт-контракты проекта (контракты токенов и мостов) должны быть покрыты автоматическими тестами;
- пользователи должны иметь доступ к сервису в любое время;
- пользователям должна быть обеспечена техническая поддержка в случае ошибок транзакций, сбоя в работе бэкенда и иных случаях;
- сервис должен интегрироваться с наиболее популярными электронными крипто-кошельками, такими как, например, MetaMask[28] или WalletConnect[29].

3.3 Выбор сетей для осуществления кроссчейн обмена

Напомним, что первоначально токен WISH был размещен в сети Binance, что является проблемой. Решить ее было предложено построением мостов

в другие сети блокчейна. Так как сеть Ethereum на сегодняшний день является самой распространенной платформой для развертывания DeFi-приложений, было бы, как минимум, странно не обратить на нее внимание. Поэтому очевидным является решение соединить сети Binance и Ethereum. Кроме того, не стоит забывать, что в свое время компанией Binance был выпущен еще один блокчейн (Binance Smart Chain), сохраняющий такую же высокую пропускную способность, что и Binance Chain, а также обладающий совместимостью с Ethereum Virtual Machine (EVM)[30]. То есть в сети BSC не просто можно писать смарт-контракты, их легко можно переносить из сети Ethereum в BSC. Ведь совместимость с EVM означает поддержку сетью BSC большинства инструментов сети Ethereum и расположенных в ней децентрализованных приложений.

Таким образом, было принято решение связать между собой следующие блокчейн-сети: Binance Chain, Binance Smart Chain и Ethereum. В перспективе планируется присоединение и других сетей.

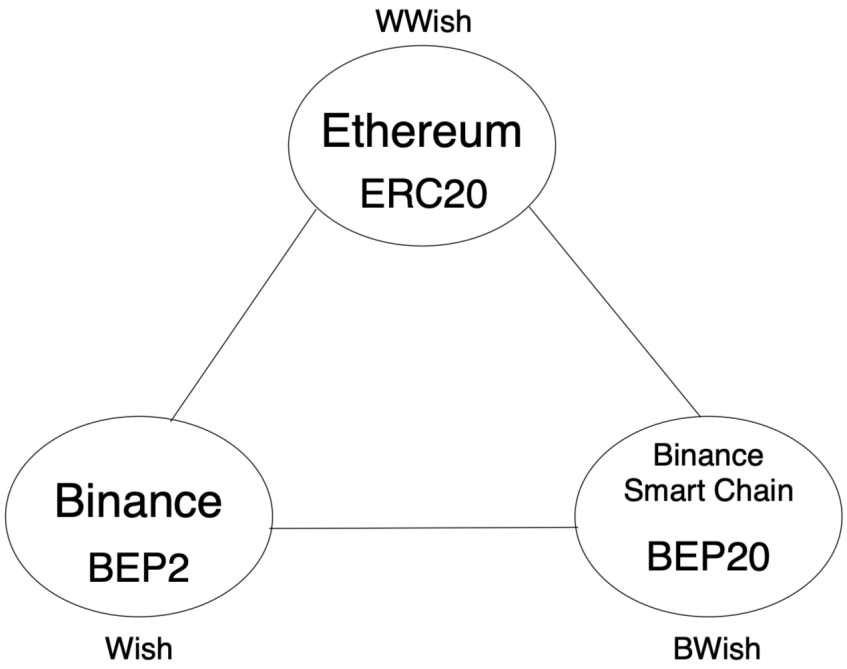

Рис. 3: Схема интегрируемых сетей.

Глава 4. Реализация проекта

В данной главе будет подробно рассмотрен функционал каждой из составляющих нашего проекта, а также их реализация.

4.1 Написание кода смарт-контрактов

Как уже было сказано ранее, проблему кроссчейн-взаимодействия предлагается решить, написав смарт-контракты аналогов токена WISH, а также контракты мостов. Смарт-контракт токена необходимо было реализовать для того, чтобы создать эквивалент токена WISH в другой сети. То есть пользователь, желающий перевести свои активы из одной сети в другую, получит не токен WISH, а его эквивалент в другой сети. Но пользователи не будут взаимодействовать с этим контрактом в нашем сервисе. Используя решение Wish Swap, пользователи будут взаимодействовать с контрактами-мостами, основной задачей которых является осуществление корректного перераспределения средств между аккаунтами.

Для реализации кода смарт-контрактов по нескольким причинам был выбран язык программирования Solidity[5]. Во-первых, Solidity является наиболее популярным языком для написания смарт-контрактов, благодаря своей хорошей документированности и коммьюнити, в которое входят и сами разработчики платформы Ethereum. Во-вторых, Solidity является кроссплатформенным, а это очень важный фактор для нас, учитывая суть проекта. И в-третьих, на языке Solidity уже написано немалое количество общедоступных библиотек. Их использование делает возможным написание смарт-контрактов в более короткие сроки, а также помогает избежать определенного рода ошибок. Ведь намного проще и правильней использовать в своем коде уже не один раз проверенные стандартизированные шаблоны, а не писать все с нуля, рискуя пропустить какую-либо важную деталь.

Итак, перечислим поля и методы, которые должен содержать код контрактов токенов и мостов.

Смарт-контракт аналога токена WISH, как уже было сказано ранее, должен наследоваться от контракта стандартного токена ERC20 из библио-

теки OpenZeppelin[26] и иметь все доступные его поля и методы. А именно:

- метод allowance, позволяющий смотреть сумму токенов, находящуюся в собственности пользователя и одобренную какому-либо контракту или адресу для распоряжения ей;
- метод balanceOf, позволяющий узнавать баланс токенов, находящихся на конкретном адресе;
- метод decimals, показывающий количество десятичных знаков после запятой (токен, как, например, один рубль содержит в себе сто копеек, состоит из 10 в степени decimals таких "копеек"). Классически, значение decimals устанавливается равным 18 по аналогии с Эфиром (ETH) – внутренней криптовалютой блокчейна Ethereum. Ведь в одном Эфире содержится 10^{18} wei (наименьшая неделимая частица Эфира);
- метод name, позволяющий узнать название токена;
- метод symbol, показывающий краткое название токена;
- метод totalSupply, позволяющий узнать количество уже выпущенных на данный момент токенов;
- метод approve, позволяющий владельцам токенов подписывать разрешение на распоряжение своими токенами для другого адреса (данный метод чаще всего используется для того, чтобы разрешать смарт-контрактам управлять определенной суммой своих токенов);
- методы decreaseAllowance/increaseAllowance, позволяющие понизить или повысить значение allowance для определенного адреса на конкретное значение;
- метод mint, позволяющий выпускать токены;
- методы transfer/transferFrom, позволяющие передавать токены между кошельками;

- соответствующие приватные поля, структуры, мэппинги, хранящие необходимую для вывода информацию (name, decimals, symbol, balanceOf и так далее).

Помимо стандартных методов и полей контракт токена WISH должен наследоваться от контракта Ownable из библиотеки OpenZeppelin и иметь метод owner (показывает адрес владельца контракта, записанный в приватном поле), модификатор onlyOwner (позволяет разрешать вызывать определенные функции только владельцу контракта), метод renounceOwnership (оставляет контракт без владельца и удаляет методы, доступные только владельцу). Наследование от данного контракта позволяет ограничивать функционал, не предназначенный для пользователей.

Кроме прочих, контракт токена будет содержать и собственные атрибуты:

- поле maxTotalSupply, содержащее цифру, показывающую максимально возможное количество выпущенных токенов (задача данного поля – ограничение функции mint проверкой значения totalSupply);

- конструктор, инициализирующий контракт и присваивающий переменным name, symbol, decimals и maxTotalSupply необходимые значения;

- пользовательский метод mint, позволяющий чеканить токены, но с ограничением maxTotalSupply и onlyOwner.

В свою очередь, контракт-мост тоже должен будет наследоваться от контракта Ownable и использовать его функционал. Более того, контракт-мост будет использовать функционал контрактов IERC20 (интерфейс стандартного ERC20 токена, позволяющий использовать в контракте его методы) и SafeMath (оборачивает арифметические операции в Solidity в публичные функции) из библиотеки OpenZeppelin. Помимо наследуемой логики, контракт-мост будет содержать:

- поле tokenAddress, которое хранит адрес контракта токена в данной сети;

- поле feeAddress, которое хранит адрес кошелька, на который будет переводиться комиссия (так как проект коммерческий, планируется внедрение комиссий за проведение операций);
- поля numOfTotalBlockchains и numOfThisBlockchain, содержащие информацию, сколько всего блокчейнов используется в нашем проекте и номер данного блокчейна (нумерация блокчейнов необходима для логгирования, которое поможет бэкенду правильно распределять транзакции между сетями);
- мэппинг (аналог словаря в языке Solidity) feeAmountOfBlockchain, который будет ставить в соответствие номеру блокчейна размер комиссии в нем;
- два события TransferFromOtherBlockchain и TransferToOtherBlockchain, позволяющие отслеживать вызов определенных методов (см. ниже);
- конструктор, инициализирующий контракт и присваивающий вышеперечисленным полям нужные значения;
- метод transferToOtherBlockchain, который станет ключевым методом, используемым в нашем проекте. Именно через этот метод будет происходить кроссчейн обмен токенов;
- методы changeInformationAboutOtherBlockchain, changeFeeAddress, setFeeAmountOfBlockchain, позволяющие менять информацию о нумерации блокчейнов, размере комиссии или об адресе для сбора комиссий.

Деятельность данной системы контрактов в различных сетях подразумевает под собой распределение реальных цифровых активов, имеющих определенную денежную ценность, вследствие чего, как уже было написано ранее, контракты должны будут пройти проверку автоматическими тестами во избежание нежелательных ситуаций. Автоматические тесты – это программы, помогающие в тестовой или локальной сети проверить работу контрактов. Основной задачей тестирования является проверить не только

правильные сценарии, но и, так называемые, "плохие"случаи, когда что-то может пойти не по плану. Например, при недостатке средств на кошльке пользователя или при неверной информации, предоставленной пользователем. Необходимо постараться "сломать"систему, выявить недочеты для дальнейшего их исправления.

4.2 Реализация бэкенда

Для реализации бэкенда использовалась распространенная связка Django + Flask (соответственно бэкенд написан на языке Python). Основным функционалом, которым обладает бэкенд, является вызов транзакций на перевод токенов в различных сетях и хранение данных об уже осуществленных переводах. Помимо основного бэкенда в прямом понимании этого слова, на языке Python также были написаны сканеры реестров блокчейн. Сканеры необходимы для непосредственной связи блокчейн-сетей. Сканер должен просматривать события транзакций, совершенных на нашем контракте в отпределенной сети, и при обнаружении нужного события (в нашем случае событие называется TransferToOtherBlockchain) сканер должен передавать всю необходимую информацию бэкенду через брокер сообщений RabbitMQ[27].

4.3 Реализация фронтенда

Кроме смарт-контрактов, основного бэкенда и сканера сетей был реализован пользовательский интерфейс с использованием фреймворка React для языка JavaScript. Основными задачами фронтенда являются:

- авторизация пользователя в приложении;
- интерфейс с полями для транзакции, необходимыми к заполнению пользователем (из какой сети, в какую сеть, какое количество токенов пользователь хотел бы перевести и на какой адрес в целевой сети);
- интеграция с кошельками MetaMask и WalletConnect для простоты подписания транзакций;

- формирование транзакций непосредственно перевода, а также транзакций на разрешение использования нашими контрактами токенов пользователя;
- интерфейс, предоставляющий ссылки на контракты токенов и мостов в обозревателе блоков Etherscan[31];
- расчет количества токенов, которое получит пользователь в целевой сети за вычетом комиссии.

4.4 Архитектура проекта

Итоговая архитектура проекта представлена на рисунке 4:

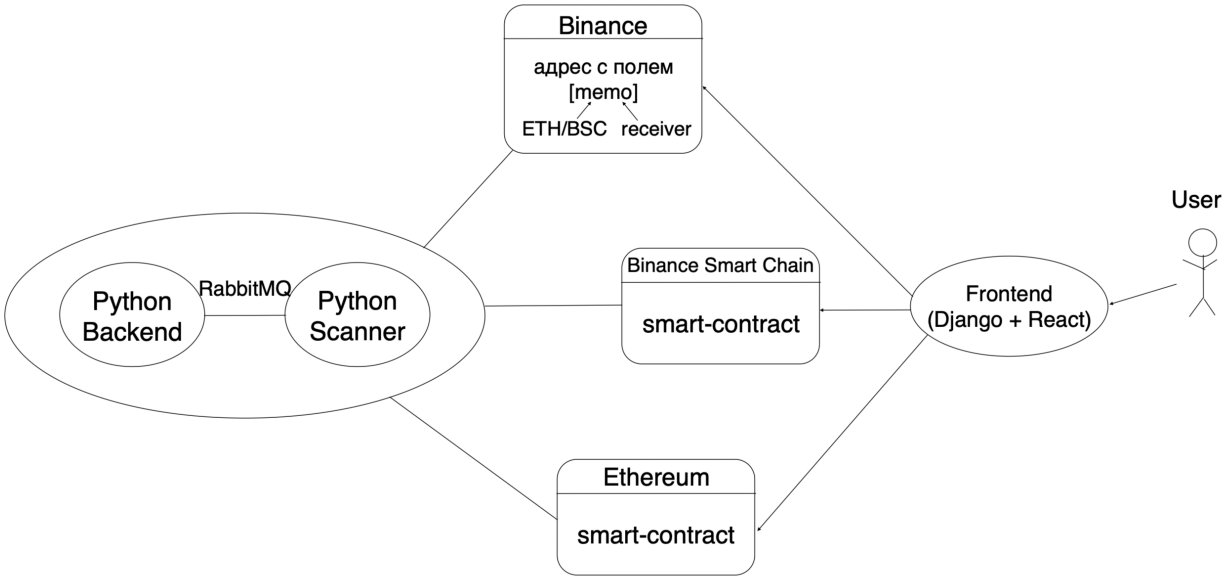

Рис. 4: Архитектура проекта Wish Swap.

4.5 Основной сценарий работы приложения

Перед тем, как описать классические шаги, через которые проходит пользователь и наша система в процессе кроссчейн-обмена, необходимо уточнить, из какого состояния наш сервис начинает свою работу. После написания всех контрактов, прохождения ими тестов в локальных и тестовых сетях наступает этап деплоя контрактов в основную сеть. Деплой

под собой подразумевает развертывание контракта в сети. При деплое один единственный раз активируется конструктор контракта, который присваивает необходимым переменным нужные значения и проводит определенные операции. В нашем случае, первым делом происходит деплой и верификация контрактов токенов. Это необходимо для того, чтобы получить адрес контракта токена и произвести деплой swap-контракта уже с этим значением. При деплое swap-контракта обязательным является следующее условие: все новые токены (WWish и BWish) в сетях Ethereum и BSC необходимо поместить на swap-контракт с целью их будущего распределения. Соответственно, при деплое swap-контракта вызывается конструктор, который записывает адрес нового токена (WWish/BWish в зависимости от сети) и вызывает функцию контракта токена mint, которая выпускает указанное количество токенов на адрес разворачиваемого контракта. Таким образом мы получаем эквивалентное количество токенов в каждой из сетей. Благодаря тому, что мы не использовали в нашем проекте протокол mint-and-burn, а продумали свою логику обмена, токен WISH не подвержен обесцениванию.

Итак, предполагаемый сценарий работы нашего кроссчейн-решения следующий:

1. Пользователь проходит авторизацию на нашем приложении (фронтенд React);

2. Пользователь подключает свой кошелек через MetaMask или WalletConnect (специальные утилиты, позволяющие безопасно подписывать транзакции собственным приватным ключом в различных приложениях, при этом не передавая закрытый ключ по сети) (фронтенд React);

3. Пользователь выбирает сети для осуществления кроссчейн-обмена (фронтенд React);

4. Пользователь заполняет поля amount (количество токенов для обмена) и receiver (адрес получателя в целевой сети) (фронтенд React);

5. Фронтенд формирует транзакции approve и transfer и предлагает поль-

зователю подписать их через MetaMask/WalletConnect, далее транзакции отправляются в блокчейн (фронтенд React, смарт-контракт в сети отправителя);

6. Сканер просматривает транзакции нашего контракта в сети отправителя и обнаруживает транзакцию на обмен токенами, передает сообщение о событии в бэкенд посредством брокера сообщений RabbitMQ вместе со всей необходимой информацией (сети отправителя и получателя, адреса отправителя и получателя, количество токенов, комиссия) (сканер Python, брокер сообщений RabbitMQ);

7. Бэкенд получает сообщение, формирует и сам подписывает транзакцию в сети получателя на перевод ему необходимой суммы токенов за вычетом установленной комиссии (комиссия за обмен взимается в токенах целевой сети). В этой же транзакции происходит и перевод комиссии на адрес feeAddress, указанный в смарт-контракте (бэкенд Python, смарт-контракт в сети получателя);

8. Токены зачисляются на адрес получателя в целевой сети.

4.6 Нюансы проекта

Несмотря на всю логичность изложения реализации проекта Wish Swap, в нем есть некоторые особенности, самая важная из которых заключается в организации процесса кроссчейн-обмена с сетью Binance Chain. При продумывании логики нашего проекта мы опирались на ее реализацию через смарт-контракты, но общеизвестен тот факт, что сеть Binance Chain не поддерживает развертывание смарт-контрактов на своей платформе. Поэтому нам пришлось искать альтернативу, которой стал обычный адрес, что является не очень хорошим решением с точки зрения безопасности (см. пункт "Безопасность решения"). Однако иных вариантов найдено не было.

Рассмотрим процесс обмена из сети Binance Chain подробнее. Он происходит следующим образом. Для того, чтобы пользователь мог отправить токены WISH из сети Binance Chain в сеть Ethereum или BSC, мы создали так называемый обменный адрес. Если пользователь захочет произвести

такой обмен, то для него, собственно, ничего не поменяется, он все также будет заполнять форму: выбирать сети, указывать количество токенов и тд., а также подписывать транзакции через электронный кошелек. Однако логика приложения чуть изменится. Ранее, при вызове функции обмена, вся указанная информация подавалась на вход соответствующей функции смарт-контракта. Здесь же, в сети Binance Chain, эта информация передается в поле memo (поле дополнительной информации, которое доступно для заполнения в сети Binance Chain). Для сети Binance Chain также был написан отдельный сканер, который работает уникальным образом из-за высокой частоты прохождения транзакций в этой сети. Задача данного сканера – отслеживать транзакции на наш обменный адрес и при обнаружении таковой отправлять сообщение бэкенду, который, в свою очередь, должен вызывать функции смарт-контрактов в сети получателя.

4.7 Безопасность проекта

Как уже было сказано ранее, хранить средства на обыкновенном кошельке является плохим тоном, так как сразу возникает вопрос безопасности и сохранности средств пользователей. Опасность заключается в том, что для доступа к вызовам транзакций обмена необходимо на бэкенде хранить приваный ключ от обменного адреса. Хранить его обычным образом в базе данных не очень хорошо, так как если базой завладеет кто-либо еще, он сможет распоряжаться токенами так, как ему заблагорассудится. Со смарт-контрактами аналогичная история: доступ к функциям перевода токенов конечным пользователям доступны лишь бэкенду, который со своего адреса вызывает эти функции. Соответственно, у этого адреса тоже есть закрытый ключ, который надо хранить. Итого, на данный момент структура бэкенда такова: мы имеем N сетей, соответственно N приватных ключей. Такая система неким образом упрощает нам задачу, так как гораздо проще придумать систему защиты для ограниченного количества приватных ключей, нежели для постоянно растущего их количества.

Итак, опишем некоторые варианты, которые позволят обезопасить пользователей нашего сервиса и их средства.

Самым очевидным решением данной проблемы является отказ от хранения приватного ключа в поле таблицы в чистом виде. Данный вариант предполагает использование полей таблицы с шифрованием данных (encrypted field / symmetric field / PGP encrypted field). Такие поля работают следующим образом: данные, сохраняемые в поле, шифруются, и не могут быть получены при отсутствии правильного пароля, пассфразы или секретного ключа, необходимого для расшифровки (такой ключ сохраняется уже отдельно, и например, заполучив базу, но не имея такого ключа, невозможно будет расшифровать поле таблицы). Достоинством такого решения является то, что теперь уже будет нельзя так просто заполучить сам закрытый ключ. Но мы все еще продолжаем хранить его в базе. Однако фактом является неизбежность этого хранения и при других вариантах защиты сервиса. Также такой метод защиты можно легко комбинировать и с другими способами усиления защищенности бэкенда, и оно практически никогда не помешает.

Еще одним вариантом решения данной проблемы является разделение бэкенда и выделение работы с приватными ключами в отдельный микросевис, который будет косвенно зависеть от основного бэкенда. Данный микросервис может быть как подсервисом основного бэкенда, так и может быть вынесен в отдельное приложение, работающее самостоятельно, возможно даже на отдельном сервере. Такой сервис обязательно должен общаться с основным бэкендом с шифрованием через HTTP/RabbitMQ канал. Также сервис должен иметь верификацию всего, что к нему приходит, на правильность отправителя. API ключ, или передаваемое секретное сообщение. Возможно даже одноразовые, по следующей схеме:

- Основной бэкенд запрашивает секретное сообщение от бэкенда приватных ключей;
- Бэкенд получает сообщение и кодирует определенным образом;
- Бэкенд передает закодированное и начальное сообщение в бэкенд приватных ключей;

- Бэкенд приватных ключей проверяет правильно ли закодировано сообщение и, если да, отдает приватный ключ.

У данного способа есть альтернатива: выделить не только хранение приватных ключей в отдельный микросервис, но и обособить весь процесс подписи транзакций бэкендом. Такой сервис должен будет получать параметры подписи от основного бэкенда, подписывать транзакцию и отдавать основному бэкенду уже подписанное сообщение. Положения об API-ключах и зашифрованных запросах из предыдущего варианта применимы и здесь.

Также не стоит забывать и о безопасности остальных частей сервиса, а не только логики бэкенда. Ведь если защитить весь бэкенд, но забыть какую-либо маленькую деталь – это может дорого нам обойтись. Из основных позиций стоит выделить:

- Обязательный HTTPS протокол;
- Обязательный firewall (как для веба сервера, так и для самого сервера);
- Жесткая политика запросов (например, Fail2Ban или иные ограничения количества запросов);
- Прокси сервера, Cloudflare;
- Изменение портов для подключения SSH;
- Использование алгоритмов RSA / ECDSA для доступа на сервер.

Глава 5. Анализ полученных результатов

На данный момент сервис уже был протестирован и запущен на основном сайте компании. Код проекта можно найти по ссылке на репозитории GitHub[32][33]. Сравним результаты работы нашего сервиса и остальных проектов, описанных в Главе 2 в виде таблицы:

Таблица 1: Анализ полученных результатов

	Wish Swap	PolkaSwap	Panama Bridge
Безопасность	да	нет	да
Реализация	да	в перспективе	да
Универсальность	нет	да	нет
Масштабирование	да	да	нет
Комиссия	100WWish/ 5BWish/ 5Wish	0.3%	0.001BNB

Для большей наглядности на Рис.5 представлены гистограммы, отображающие количественные оценки работы сравниваемых сервисов.

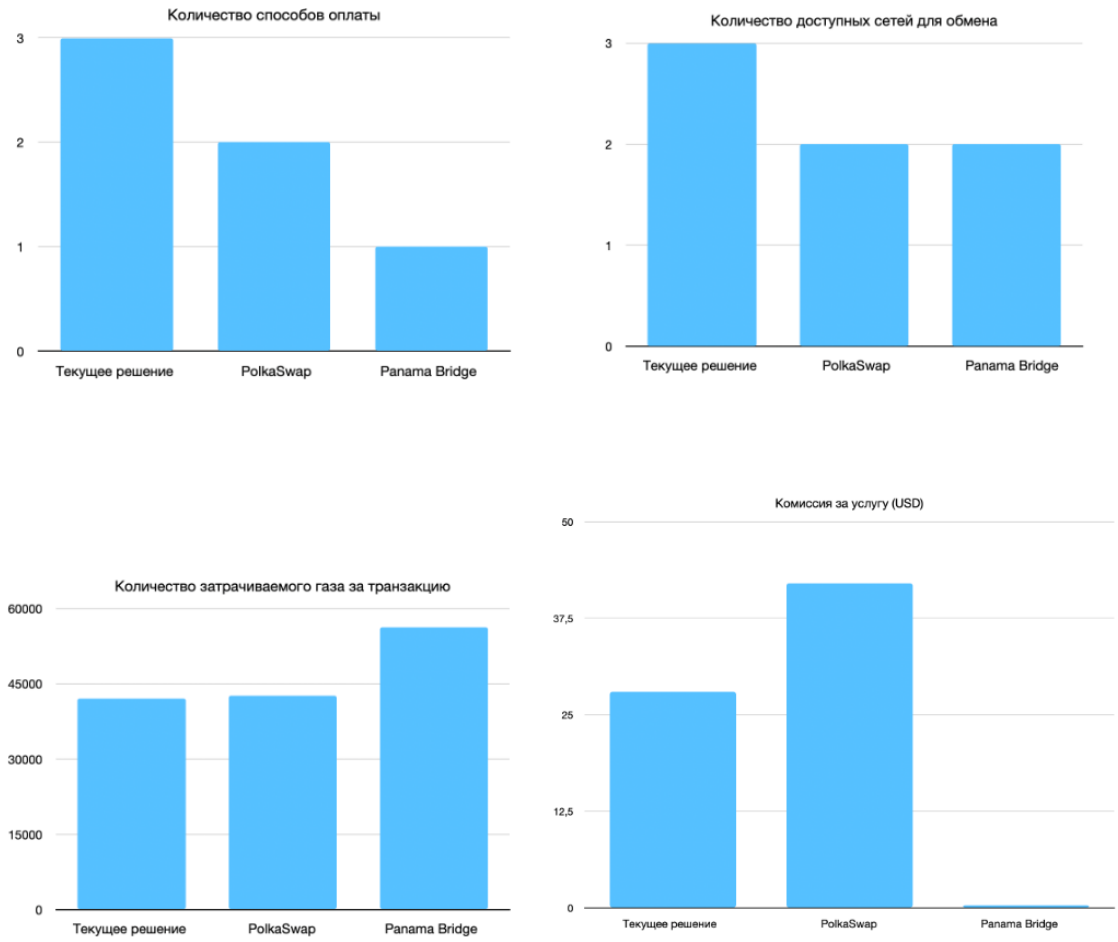

Рис. 5: Сравнение текущего решения со сторонними сервисами.

Выводы

Изучая проблему невозможности самостоятельной передачи информации между различными блокчейнами, мы пришли к тому, что это возможно реализовать с помощью блокчейн-мостов, которые по своей сути представляют слаженный механизм работы смарт-контрактов и бэкенда.

Важно заметить, что представленная концепция решения проблемы передачи конкретного токена стандарта ERC20 между различными сетями блокчейна может быть легко распространена и на другие стандарты то-

кенов (например, NFT-токенов), а также и для любой другой публичной информации, хранящейся в блокчейнах.

В дальнейшем сервис планируется развивать, подключая к нему новые сети блокчейна для осуществления кроссчейн обмена с ними.

Заключение

В ходе выполнения выпускной квалификационной работы были решены задачи и достигнуты следующие результаты:

1. подробно изучена технология блокчейн, ее основные принципы работы и внутреннее устройство;
2. рассмотрены и проанализированы уже существующие проекты, решающие поставленную проблему;
3. основываясь на анализе уже готовых проектов, а также учитывая интересы компании, сформирована идея решения заявленной задачи;
4. выдвинуты технические требования к реализации проекта;
5. подобраны инструменты, технологии и фреймворки, удовлетворяющие техническим требованиям;
6. написаны, оптимизированы и протестированы смарт-контракты мостов и токенов;
7. проведено сравнение собственного решения со сторонними решениями, изученными в главе 2;
8. выработан путь дальнейшего развития проекта.

Список литературы

[1] A. Averin, O. Averina Review of Blockchain Technology Vulnerabilities and Blockchain-System Attacks // 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon). 2019.

[2] Zetzsche D., Arner D., Buckley R.. Blockchain disruption and decentralized finance: the rise of decentralized business models // Journal of Financial Regulation. 2020. Vol. 6. Iss. 2. P. 172–203.

[3] Официальная статистика сайта DeFi Pulse, URL: `https://defipulse.com/` (дата обращения: 20.04.2021).

[4] Buterin V.. A next generation smart contract and decentralized application platform // Ethereum Whitepaper. 2013.

[5] Официальная документация Solidity, URL: `https://docs.soliditylang.org/en/v0.8.4/` (дата обращения: 21.04.2021).

[6] Официальная документация Binance, URL: `https://docs.binance.org/` (дата обращения: 11.03.2021).

[7] Tron Whitepaper, URL: `https://tron.network/static/doc/white_paper_v_2_0.pdf` (дата обращения: 11.03.2021).

[8] Gavin Wood. Polkadot: vision for a heterogenus multi-chain framework // Polkadot Whitepaper. 2016.

[9] NEO Whitepaper, URL: `https://docs.neo.org/docs/en-us/basic/whitepaper.html` (дата обращения: 11.03.2021).

[10] Cangshu Li, Yan Shen. The potential impacts and risks of global stablecoins // China Economic Journal. 2017. Vol. 14. Iss. 1. P. 39-51.

[11] Semyon Malamud, Marzena Rostek. Decentralized exchange // American Economic Review. 2017. Vol. 107. No. 11. P. 3320-3362.

[12] Официальная документация Polkadot, URL: https://wiki.polkadot.network/ (дата обращения: 11.03.2021).

[13] Wenbo Wang, Peizhao Hu, Hoang Dinh Thai. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks // IEEE Access. 2019.

[14] Официальная документация Tendermint, URL: https://docs.tendermint.com/ (дата обращения: 11.03.2021).

[15] BABE paper, URL: https://research.web3.foundation/en/latest/polkadot/block-production/Babe.html (дата обращения: 11.03.2021).

[16] Alistar Stewart, Eleftherios Kokoris-Kogia. GRANDPA: a Byzantie finality gadget. 2020.

[17] Alfonso Cevallos, Alistair Stewart. A verifiably secure and proportional committee election rule. 2020.

[18] Fan Yang, Wei Zhou, Qingqing Wu, Rui Long, Neal N. Xiong, Meiqui Zhou. Delegated Proof of Stake With Downgrade: A Secure and Efficient Blockchain Consensus Algorithm With Downgrade Mechanism. 2019.

[19] Официальный сайт Polkaswap, URL: https://polkaswap.io/ (дата обращения: 11.03.2021).

[20] Официальный сайт SORA, URL: https://sora.org/ (дата обращения: 11.03.2021).

[21] Vijay Mohan. Automated Market Makers and Decentralized Exchanges: a DeFi Primer. 2021.

[22] Drew Stone. Trustless, privacy-preserving blockchain bridges. 2021.

[23] Официальный сайт Binance Panama Bridge, URL: https://www.binance.org/en/bridge (дата обращения: 11.03.2021).

[24] Официальный сайт MyWish, URL: https://mywish.io/ (дата обращения: 11.03.2021).

[25] Shahar Somin, Goren Gordon, Alex Pentland,Erez Shmueli, Yaniv Altshuler. ERC20 Transactions over Ethereum Blockchain:Network Analysis and Predictions. 2020.

[26] Библиотека смарт-контрактов OpenZeppelin, URL: `https://openzeppelin.com/` (дата обращения: 11.03.2021).

[27] Официальная документация к брокеру сообщений RabbitMQ, URL: `https://www.rabbitmq.com/documentation.html` (дата обращения 11.03.2021).

[28] Официальный сайт кошелька MetaMask, URL: `https://metamask.io/` (дата обращения: 11.03.2021).

[29] Официальный сайт кошелька WalletConnect, URL: `https://walletconnect.org/` (дата обращения: 11.03.2021).

[30] Everett Hildenbrandt, Manasvi Saxena, Nishant Rodrigues, Xiaoran Zhu, Philip Daian, Dwight Guth. KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine // 2018 IEEE 31st Computer Security Foundations Symposium (CSF). 2018.

[31] Открытый реестр транзакций сети Ethereum, URL: `https://etherscan.io/` (дата обращения: 11.03.2021).

[32] Ссылка на репозиторий GitHub (бэкенд) проекта Wish Swap, URL: `https://github.com/MyWishPlatform/crosschain_swap_backend/` (дата обращения: 16.04.2021).

[33] Ссылка на репозиторий GitHub (смарт-контракты) проекта Wish Swap, URL: `https://github.com/RitaTcepeleva/bridge-contracts` (дата обращения: 19.04.2021).