Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт математики, физики, информатики и технологии

Кафедра информатики, информационных технологий и методики обучения информатике

Веб-сервис для управления цифровыми активами

Выпускная квалификационная работа

Квалификационная работа
допущена к защите
зав.кафедрой
«____»_____20___г

_____
(подпись)

Исполнитель:
Микушин Даниил Андреевич
Обучающийся группы ПИВС-15.01z

_____
(подпись)

Руководитель:
Алексеевский Петр Иванович,
Старший преподаватель кафедры ИИТиМОИ

_____
(подпись)

Екатеринбург 2020

Содержание

ВВЕДЕНИЕ

Актуальность темы. В настоящее время в ряде значительных стран обретает значимость понятие «цифровая экономика».

Развитие цифровых технологий и большой охват компьютерных сетей, позволяет создавать новые революционные решения в экономико-финансовых областях. Государства, предприятия и другие участники рынка, которые придерживаются современных инструментов, в кратчайшие сроки приобретают новых клиентов, выгодные условия сделок, увеличивают конкурентоспособность и эффективность рабочих процессов. [25]

Цифровые активы, представляющие собой криптовалюты, созданные с применением блокчейн-технологии, применяются достаточно часто в финансовых отношениях. Сложившееся ситуация на рынке, заставляет обратить своё внимание на необычный вид активов, даже самых больших гигантов классической финансовой системы. Изучение, применение или создание собственных инструментов для управления такими цифровыми активами, становится основной деятельностью многих учреждений. [22]

Тема представляет теоретический и практический интересы, потому что необходимо быть в курсе современных разработок, а также уметь на практике применять и создавать различные решения в своей профессиональной деятельности, чтобы оставаться востребованным специалистом на рынке.

Цель исследования изучить современные сервисы и платформы, позволяющие управлять цифровыми активами, созданные с применением блокчейн технологии, а также получить знания и опыт в создании собственного решения способное удовлетворить нужды бизнеса.

Объектом исследования являются проблемы, возникающие при управлении цифровыми активами, представленных в виде криптовалют, с использованием существующих программных решений. А также различия между способами взаимодействиями у одиночных и бизнес пользователей.

Предмет исследования. Блокчейн сети, их сходства и особенности. Программная реализация собственного сервиса для удовлетворения потребностей бизнес пользователей в управлении цифровыми активами.

Задачи исследования:

- Собрать и проанализировать информацию о готовых решениях в сфере управления цифровыми активами, представленных в виде криптовалют;
- Выбор программных средств для создания собственного решения;
- Разработка программного продукта;
- Внедрение и эксплуатация.

Методы исследования:

- Теоретические методы: анализ, классификация, формализация;
- Эмпирические методы: сравнение, описание.

Теоретическая значимость заключается в сборе актуальной информации по современным способам обмена финансово ценными ресурсами. Изучение и анализ применяемых технологий и сервисов с кратким описанием их возможностей.

Практическая значимость: получение опыта в сетевом программировании. Работающий продукт готовый к использованию и дальнейшим модификациям. Возможность развивать и использовать проект на коммерческой основе.

ГЛАВА 1. ЦИФРОВЫЕ АКТИВЫ, ТЕХНОЛОГИЯ БЛОКЧЕЙН И СЕРВИСЫ ДЛЯ УПРАВЛЕНИЯ ЦИФРОВЫМИ АКТИВАМИ.

1.1 Цифровые активы и их управление

Чтобы дать определение цифровому активу, обратимся к словарю банковских терминов, официально закреплённых и используемых в экономических отношениях и разрешении правовых споров.

Цифровой актив — информационный ресурс, производный от права на ценность и обращающийся в распределенном реестре в виде уникального идентификатора. Цифровой актив является информационным ресурсом в том смысле, что представленная в цифровом виде информация о ценности обладает такими основными свойствами информационного ресурса, как:

1. информация структурирована по определенным параметрам и категориям;
2. информация фиксируется на цифровом носителе;
3. информацию можно хранить, передавать, обменивать, использовать и т. п.

Ценностная составляющая в контексте определения понятия «цифровой актив» представлена в сфере материальных и нематериальных благ компонентой «Ценность». Цифровой актив является своего рода гарантированным правом претендовать на определённое значение стоимости (на ценность), заложенное в данном цифровом активе. Другими словами, цифровой актив выступает цифровым отображением ценности. [24]

В настоящий момент приравнивание криптовалют к цифровым активам в процессе утверждения, несмотря на это суды приравнивают криптовалюты к цифровым активам или к другому имуществу. В данной работе под цифровыми активами понимаются криптовалюты, созданные на основе технологии блокчейн.

Под управлением понимается, возможность своевременно получать количественную информацию об остатках или новых поступлениях. Иметь возможность совершить операцию перераспределения средств между собственными кошельками, а также полное или частичное списание в пользу другого участника финансовых отношений. Другими словами управление цифровыми активами позволяет полностью контролировать состояние этих средств и своевременно совершать различные операции обмена.

1.2 Описание технологии Блокчейн

Блокчейн (от английского blockchain), формально это растущий список записей, называемые блоками, которые используют криптографию для обеспечения связей между собой. Каждый блок содержит криптографическую хеш-сумму предыдущего блока, временную метку и данные транзакции. Данную технологию можно представить в виде алгоритмической структуры данных, как «Хеш-дерево» или дерево Меркла (Merkle tree). Одиночную цепочку блоков можно представить, в виде связного списка (linked list), где связью будет хеш-сумма предыдущего блока. [16]

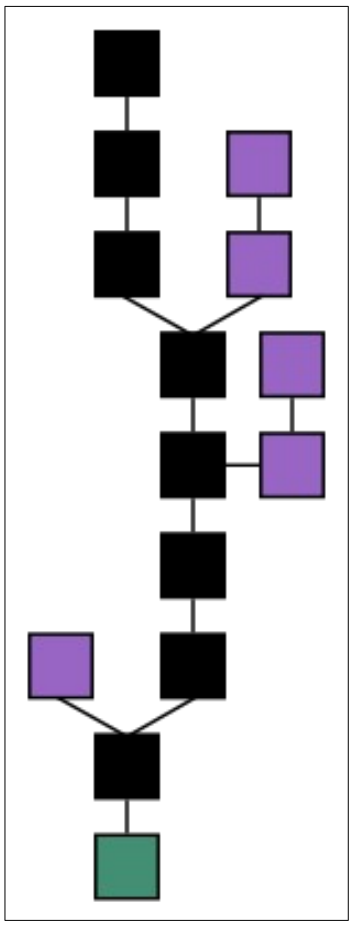

Рисунок 1. Схематичное представление Блокчейн.

На рисунке 1, изображено схематичное представление цепочки блоков. Чёрным цветов изображена основная цепочка, которая содержит самую длинную серию блоков от первоначального блока (genesis), отмечен зелёным цветом. Фиолетовым цветом отмечены «осиротевшие» блоки, которые не получили достаточной длинны блоков, чтобы стать основной цепочкой.

Блокчейн изначально проектировался невосприимчивым к изменению зафиксированных и подтверждённых в цепи данных. За счёт хранения хеш-суммы предыдущего блока, малейшее изменение данных, заставит переписывать все последующие блоки цепи. Помимо этого целостность гарантируется и проверяется всеми участниками распределённой сети. Блокчейн можно использовать в качестве распределенного регистра данных или

как общую бухгалтерскую книгу, тогда все участники сети верифицируют новые блоки, рассчитывают хеш-сумму предыдущего блока и добавляют блок к основной цепочке. Если будет попытка подменить данные, то злоумышленник должен обладать 51% вычислительных мощностей всей сети, только тогда он сможет прикрепить неправильные данные к цепочке.

Также необходимо разделять технологию Блокчейн и основанные на ней криптовалюты, которые являются лишь частным применением технологии и не являются тождественными понятиями. [9]

1.3 История развития Блокчейн и его виды.

Блокчейн технология изначально разрабатывалась для запуска криптовалюты Bitcoin и её появление на свет приравнивают к дате публикации статьи «Bitcoin: A Peer-to-Peer Electronic Cash System», 31 октября 2008 года. В настоящее время нет достоверных сведений, кто входил в состав разработчиков проекта, кроме псевдонима человека или группы людей - Сатоси Накомото (Satoshi Nakamoto). [15]

С 2010 года Сатоси не принимает участие в работе сети Bitcoin и не занимается внесением изменений в структуру Блокчейна. Независимые сообщества и коммерческие организации, начали модифицировать технологию, создавая свои варианты блокчейна и криптовалюты на их основе. Так в 2015 году была запущена блокчейн платформа Ethereum, которая содержала иную структуру данных и новые возможности работы в сети.

Классификацию блокчейн считают условной, т. к. несмотря отличия в работе и предоставляемые возможности технология остаётся прежней.

Виталий Бутерин, создатель Ethereum в своей статье сделал классификацию на три вида по контролируемости:

1. Публичный блокчейн (public blockchain), каждый участник может участвовать в согласовании принятия решений, платёжные транзакции никем не контролируются;
2. Консорциум (consortium blockchain) — согласование решений контролируется отдельными, избранными участниками;
3. Частный блокчейн (fully private blockchain) — все транзакции и согласования, отслеживаются и контролируются централизованным участником.

В России упрощенно классифицируют на два вида, открытые и закрытые [25].

1.4 Варианты применения блокчейн технологии.

Криптовалюты - самое популярное применение блокчейн технологии. Созданные на основе блокчейн распределенные книги с финансовыми записями, позволили создать абсолютно новый ценный актив, существующий в цифровом виде. Если криптовалюта строится независимо от корпораций, то никто не сможет отменить перевод средств или изменить состав кошельков. Это именно тот момент который сделал криптовалюты востребованными.

Подтверждение владения или авторство произведения, один из способов доказать авторство над своим аудио файлом, изображением, видео файлом, это записать информацию о нём в децентрализованный блокчейн, где эта информация останется навсегда в неизменном состоянии.

Средства электронного голосования, с помощью блокчейн технологии есть возможность создать платформу для осуществления анонимных онлайн голосований.

Азартные игры и обмен цифровыми активами, на базе блокчейн и открытого исходного кода, можно создают прозрачные и честные платформы для проведения лотерей, азартных игр или места для обмена цифровыми

активами, которые будут необратимы и происходят по ожидаемому сценарию. [11]

Цифровая идентификация и ограничение распространения персональных данных. Используя блокчейн, можно создавать и распространять свои идентификационные данные. Также можно ограничивать доступ к такой информации, данные будут зашифрованы в открытой базе данные и предоставляться только с разрешением владельца записи.

В 2014-2018 годах на фоне резко возросшей популярности криптовалют и блокчейн, сообщества внедряли технологию, буквально, во все области, даже в которых они были лишней нагрузкой на бизнес и общую информационную систему. Так создавались торговые площадки, в которых использовались собственно выпущенные цифровые токены, вместо традиционных денег. По сути использовались лишь для выпуска собственной валюты, а сам бизнес проходил по привычной схеме. Эти и многие другие действия отрицательно сказались на популярности криптовалют и технологии в целом.

1.5 Криптовалюты, их виды и особенности

Криптовалютой принято считать цифровой актив, который использует для существования платформу, основанную на криптографических методах, например блокчейн. В настояще время было создано большое множество различных криптовалют, о большинстве из них не было известно шире ближайшего круга общения автора этой валюты. Множество разработок основываются на первой созданной криптовалюте Bitcoin, немного модифицируя исходный код, внося новые отличительные функциональные черты, криптовалюта считается отдельным ответвлением и развивается самостоятельно. Но также существуют и другие реализации, отличающиеся пользовательским функционал, политикой работы, надёжности использования.

1.5.1 Bitcoin и его ответвления

Bitcoin остаётся криптовалютой с самой высокой стоимостью на биржах, несмотря на ограниченный функционал и медлительную работу. Данная криптовалюта использует консенсус «Доказательство работы» (Proof-of-work), который обязывает участников решать сложные криптографические задачи, чтобы создать новый блок. Новый блок содержит в себе новые платёжные транзакции, помимо этого блок должен содержать «красивую» хеш-сумму всех транзакций, хеш предыдущего блока и случайное число. «Красота» хеша определяется путём количества нулей в начале этого хеша. Участник сети (майнер), который первым смог собрать такой блок и отправить в сеть, получит вознаграждение за проделанную работу. Чтобы поддерживать частоту появления новых блоков в среднем 1 блок в 10 минут, после 2016 блоков, происходит перерасчёт сложности, т. е. переопределение «красоты» хеша. Также, каждые 210000 блоков, уменьшается количество вознаграждаемой криптовалюты, чтобы поддерживать свою стоимость. Вся валюта управляется автономно и все участники следуют протоколу, в собственных интересах. [23]

Многих пользователей не устраивает частота появления новых блоков, т. к. фиксирование оплаты может произойти от 5 минут до нескольких часов или дней. Других пользователей не устраивает применяемые нововведения в сеть. В виду того что исходный код проекта был открыт и описание механизмов работы протокола изначально доступны общественности, сообщества могли незатратно создавать собственные реализации. Такие криптовалюты могут слабо отличаться механизмом работы, а иногда и просто идеологическим использованием.

Перечень самых крупных Bitcoin подобных валют с их особенностями:

Litecoin — создан для упрощения создания новых блоков, тем самым ускоряя подтверждение платёжных транзакций. Скорость появления новых блоков в среднем 1 блок в 2,5 минуты.

Bitcoin Cash — прямое ответвление (fork) от Bitcoin, был создан, чтобы увеличить максимальный размер блока до 8 МБ против 1 МБ.

Dogecoin — основанный на Litecoin, чтобы отдалится от отрицательной репутации связанной с Bitcoin, в частности продажа наркотиков. Сообщество неоднократно проводили и поддерживали сборы средств на благотворительность. Имеет фиксированную комиссию при переводе средств и неограниченную эмиссию новых монет.

Dash — первая полностью анонимная валюта, изначально имела название DarkCoin. В этой криптовалюте используются анонимайзеры, делящие транзакции на более мелкие части и смешиваются между собой, не позволяя отследить прямого отправителя и получателя.

Эти и другие подобные ответвления от Bitcoin в целом имеют те же проблемы и ограничения первоначальной платформы. Проблемы устраняются за счёт создания нового ответвления, но протоколы взаимодействия остаются прежними, что позволяет использовать схожие практики на клиентской стороне.

1.5.2 Криптовалюты и платформы

Среди обычных реализаций электронных денег существуют более сложные реализации, такие как Ethereum и Waves. Помимо простых возможностей передачи цифровых активов между кошельками, они могут создавать и выпускать собственные монеты (токены), для нужд пользователя.

Waves — это блокчейн платформа, созданная для облегчения выпуска собственных цифровых монет, а также для организации краудфандинговых кампаний. В отличии от криптовалюты Bitcoin, Waves использует консенсус «Доказательство доли владения» (Proof-of-stake), при таком методе работы нет необходимости владеть большими вычислительными мощностями. Новый блок будет формироваться с большей вероятностью аккаунтом, который обладает наиболее большим балансом. Данная платформа была разработана предпринимателем Александром Ивановым в 2016 году. По своей идее платформа позволяет людям без знаний в программировании и криптографии создать собственную кампанию для привлечения цифровых средств, которые должны пойти на реализацию будущего бизнеса. [29]

Ethereum — блокчейн платформа, предназначенная для создания собственных сервисов, используя блокчейн и умные контракты (smart contracts). Умные контракты представлены в виде общих классов, реализуемые на любых языках программирования и компилируются в исполняемый код. Данная платформа, также позволяет выпускает собственные токены, но благодаря умным контрактам имеется возможность добавить свои особенности передачи, получения или эмиссии токенов. В данным момент криптовалюта использует Proof-of-work консенсус, но планируется переход к Proof-of-stake, чтобы сократить большое количество потребляемой энергии.

Подобные платформы, позволили многим организациям организовать первоначальное привлечение средств для реализации своих идей и дальнейшего построения бизнеса. Также это породило множество мошеннических проектов, которые закрывались сразу после получения средств. [30]

1.5.3 Криптовалюты с уникальными особенностями

В данном разделе представлены наиболее крупные криптовалюты, которые создавались отдельно от Bitcoin и имеют уникальные особенности.

Zcash — криптовалюта, которая позволяет скрыть отправителя, получателя, а также сумму перевода, несмотря на то, что все платежи записываются в общедоступный блокчейн. Это первая криптовалюта, которая использует криптографический протокол «Доказательство с нулевым разглашением» (Zero-knowledge proof), позволяющая делать переводы, полностью или частично приватными. [37]

Ripple — основан на приватном блокчейн, представляет собой консенсусный реестр, а верификацией транзакций занимается сеть независимых распределенных серверов. Помимо собственной валюты, данная система позволяет совершать межбанковские переводы, в различных валютах и разные страны. [36]

Monero — криптовалюта основывающееся на протоколе Cryptonight, а также обфусцирует данные внутри блокчейна, что делает переводы анонимными. Использование протокола Cryptonight, позволяет осуществлять генерирование новых блоков на процессорах и на видеокартах, это даёт возможность включаться в сеть без больших вложений в оборудование. [35]

1.6 Сервисы и площадки для управления цифровыми активами

Чтобы совершать обменные операции в системах построенных на блокчейн, необходимо иметь собственный адрес, на который будет зачислятся средства или списываться. Для подтверждения владения адресом необходимо иметь приватный ключ от этого адреса. В схожих с Bitcoin криптовалютах адрес — это хеш-сумма публичного ключа, выведенного из приватного, хеш - сумма проходит через функции SHA256, а затем RIPEMD160. Для получения средств участие владельца адреса не требуется, а вот для перевода необходимо подписать транзакцию криптографическим ключом.

Также необходимо иметь доступ в децентрализованную сеть, чтобы отправить платёжную транзакцию, для этого можно запустить собственный узел, который будет полной репликой всей сети. Другим вариантом будет использование узла открытого для общего доступа.

Зачастую многие сервисы отличается способами взаимодействия с узлами, наличию ключей у пользователя и количеством поддерживаемых криптовалют. Клиентские реализации можно разделить на два вида:

1. Толстые — требуют загрузку блокчейн сети на клиентскую машину, чем занимают множество места на диске и времени на подготовку к работе.

2. Тонкие — соединяются с удалёнными серверами сервиса, как правило по HTTP протоколу, которые сервисом передаются в узлы сети.

1.6.1 Примеры клиентов и их особенности

Bitcoin Core, Ethereum Wallet — это официальные клиенты представленные сообществами Bitcoin и Ethereum, требуют загрузку блокчейн сети, ключи хранятся локально. Предоставляют полную свободу в получении любой доступной информации по адресам и их балансам, а также в проведении обменных операций. Для каждой отдельной криптовалюты необходимо держать отдельный клиент и переключаться между разными кошельками. [31,32]

Exodus, Jaxx — тонкие мультивалютные кошельки, помимо привычных криптовалют поддерживают токены стандарта ERC20. Ключи от кошельков можно импортировать между платформами, а также восстанавливать из mnemonic фразы, такие как BIP39. Подобные сервисы используют свои узлы сети для функционирования, поэтому могут собирать информацию о пользователях, их достатке и направлении денежных потоков. Позволяют переключаться между кошельками, но одновременно с одного клиента можно управлять только одним кошельком. [33, 34]

Blockchain Wallet, Coinbase — web клиенты, позволяют пользоваться криптовалютными кошельками, как обычным интернет баком, вся сложность

взаимодействия с блокчейн скрыта от пользователя. В стандартном режиме работы, не предоставляют пользователям приватные ключи от кошельков. Сервисы подобного типа, представляются условно бесплатными, либо могут взимать дополнительную комиссию за совершение переводов, также собирают пользовательские данные. Ограничены по количеству валют в кошельках, предоставляют ограниченный WEB API. [27, 28]

Рисунок 2. Пример пользовательского интерфейса клиента Exodus

1.6.2 Сценарии взаимодействия разных типов клиентов

При взаимодействии в схеме, когда одиночный клиент совершает платёжную операцию в пользу бизнес клиента или наоборот ожидает получения средств, то не проявляется никаких сложностей в управлении кошельками и их балансами. Достаточно использовать любой клиент или web сервис, в зависимости от целей сохранения приватности данных и желаемой валюты. Со стороны бизнеса, необходимо вести сразу несколько кошельков, в различных валютах, а также необходимо автоматизировать получение средств и осуществлять операции сразу по нескольким групп кошельков. Если использовать существующие программные средства, то процесс обмена и

актуализации информации, полностью ложиться на пользователя этого ПО, которому придётся вручную переключаться между кошельками, собирать и сводить всю динамически изменяющуюся информацию.

Если взаимодействие происходит между двумя бизнес клиентами, то проблема групповых операций становится более актуальной, необходимо с нескольких кошельков сделать перевод на один адрес или несколько адресов, а также моментально отреагировать на поступление средств, для своевременного начала выполнения обязательств со своей стороны. Таким образом сервисы и инструменты предназначенные для использования одиночными пользователями, плохо подходят в схеме, когда возрастает частота операций и становится критично время реакции на изменение данных.

Выводы по первой главе.

В первой были описаны основные моменты технологии блокчейн. Представлена краткая история развития и приведены примеры возможных применений технологии вне экономической сферы. Также перечисленны самые известные, функционирующие на текущий момент, криптовалюты с указанием их отличий друг от друга. В дополнении к этому представлены клиентские приложения и сервисы, для взаимодействия с блокчейн сетью и управления цифровыми активами. Описаны способы взаимодействия различными типами клиентов.

Полученная информация и результаты её анализа помогут в дальнейшем реализовать проект, который сможет решить проблемы бизнес клиентов в управлении своими цифровыми активами. Очень важно знать достоинства и недостатки существующих систем, чтобы получить максимально качественный продукт.

ГЛАВА 2. ПОСТАНОВКА ПРОБЛЕМЫ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РЕШЕНИЯ.

2.1 Технические особенности работы блокчейн в Bitcoin

Bitcoin первая криптовалюта и при её реализации было добавлено несколько особенностей, которые на протяжении длительной работы привели к сложностям в использовании и избыточному потреблению ресурсов. Другие ответвления от Bitcoin частично исправляют эти проблемы, но лежащие в основе недостатки, остаются в системе.

В системе Bitcoin платёжные транзакции объединяются в блоки, блоки связываются друг с другом, образуя сеть, которая сохраняет целостность данных, просто по структуре хранения этих данных.

Каждая новая транзакция создаёт в системе «Не потраченную» (Unspent) транзакцию, независимо от того что это за транзакция, перевод с адреса на адрес или вознаграждение за создание нового блока. В последствии не потраченные транзакции можно использовать в переводах на другие адреса, тем самым создавая новые. В случае с вознаграждением за новый блок, происходит эмиссия монет, а не использование Unspend транзакции. Чтобы совершить перевод, необходимо:

- Наличие у адреса или нескольких адресов поступления средств, т. е. не потраченные транзакции;

- Создать новую транзакцию, которая будет помещена в новый блок системы;

- Указать на входах идентификаторы не потраченных транзакций вместе с их порядковым номером «выхода» транзакции

- Указать, как минимум один «выход» новой транзакции, который станет новой не потраченной транзакцией в системе

- Подписать все входы, соответствующими ключами, адреса которых были использованы.

Благородя тому, что система в новых транзакциях заставляет использовать предыдущие транзакции, будет невозможным сделать перевод средств с адреса, если в него не было зачислений. Если сравнить с банковской системой где, у пользователя есть единый баланс и во время траты средств системе не важно, какие конкретно используются средства, то в системе Bitcoin и его подобиях, необходимо прямо указывать какая именно платёжная операция была потрачена.

<table>
<tr><td colspan="5">Транзакция 1</td></tr>
<tr><td colspan="2">Входы</td><td colspan="3">Выходы</td></tr>
<tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5">Комиссия: 7</td></tr>
</table>

Рисунок 3. Схематичное представление транзакции.

На рисунке 3 представлено схематичное представление транзакции в системе Bitcoin и в подобных ему системах. Пояснения к обозначениям транзакции:

1 – Идентификатор (ID) созданной транзакции;

2 – Порядковый номер входа, в текущей транзакции;

3 – ID транзакции + N (номер) использованного выхода «ID_N»;

4 – Порядковый номер выхода (N);

5 – Сумма выхода;

6 – Адрес выхода (получателя);

7 – Сумма комиссии, является разницей между суммой входов и суммой выходов;

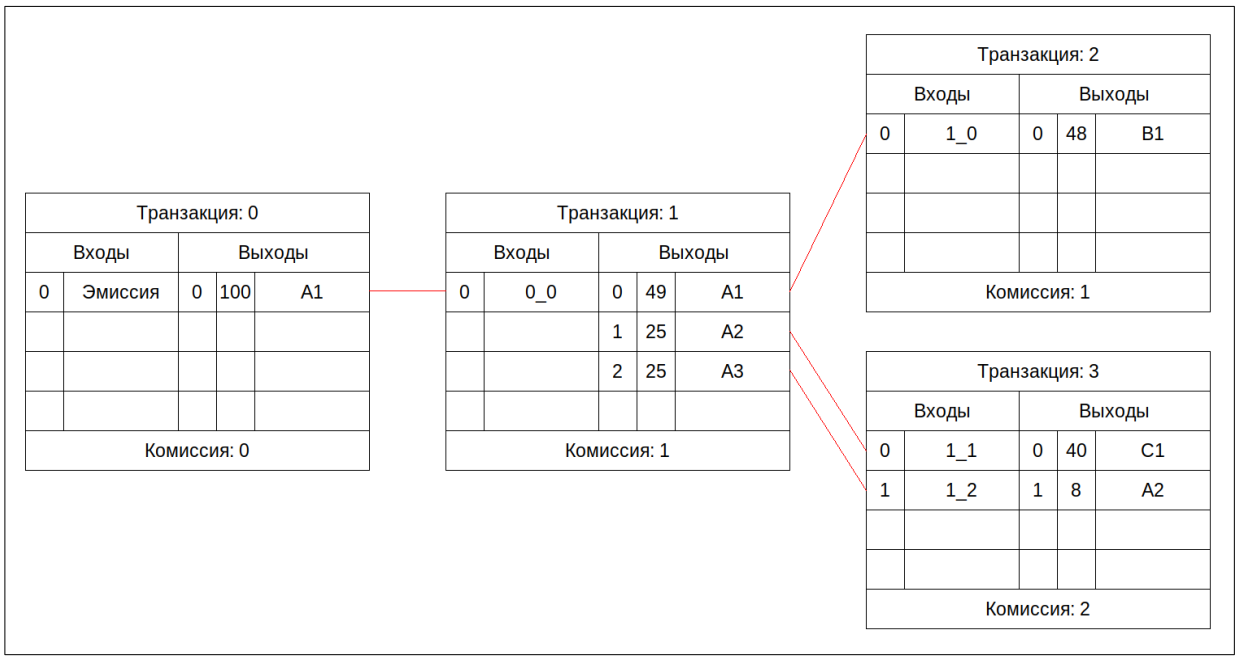

Рисунок 4. Пример движения транзакций в сети.

На рисунке 4 представлен возможный вариант движения средств в Bitcoin подобной сети. В транзакции 0, средства появились за счёт эмиссии и были зачислены на адрес A1. В результате этого в системе появилась не потраченная транзакция 0_0 (ID: 0, N: 0). Далее данный выход был использован в транзакции 1, сумма транзакции разделилась на 3 части: по 25, 25 и 49, неиспользованный остаток, ушёл в комиссию перевода. Транзакция 1, создала в системе 3 не потраченных транзакции, у адресов A1, A2, A3. Перевод на адрес A1, с адреса A1, можно считать сдачей от операции, т. к. перечислен самому себе.

После проведения транзакций 2 и 3 в системе появились не потраченные транзакции: 2_0 у адреса B1, 3_0 у адреса C1, 3_1 у адреса A2. В транзакции 3 на входах указаны две транзакции от двух разных адресов, также можно комбинировать различные транзакции с одного адреса. Транзакцию 3_1 можно считать сдачей, также здесь была поднята комиссия за перевод, в большинстве криптовалют, это повышает шанс появиться в новом блоке.

Изучая пример схемы движения транзакций достаточно просто увидеть различия между традиционной банковской системой, где платёжная операция является лишь изменением одного числа, баланса пользователя.

2.2 Описание возможных проблем и ограничений клиентских программ

В следствии того, что для совершения платёжных операций нам необходимо знать предыдущие транзакции, возникает задача хранения этих данных. Если использовать полный узел сети, в котором находится вся информация о блокчейн сети, то можно пройтись по всем блокам и отыскать нужные транзакции. Но т. к. история блоков очень большая, на это может потребоваться очень длительное время, поэтому клиентские узлы могут дополнительно хранить список не потраченных транзакций, отдельно от основной информации, это называется «кэшем не потраченных транзакций». Клиентские программы, которые загружают полную сеть на устройство пользователя, может отдельно собирать не потраченные транзакции только адресов, которые создаёт пользователь, чтобы иметь возможность быстро подсчитать баланс кошелька или произвести списание средств.

Если пользователь решит использовать клиент, который загружает всю блокчейн сеть, возникнут следующие проблемы и ограничения:

1. Загружаемая сеть занимает очень много места на устройстве. На 2019 год полная сеть занимает 210 Гб.

2. Полная синхронизация до актуального состояния, потребует много времени, которое напрямую зависит от мощности клиентского оборудования. В среднем, если выделить на процесс синхронизации 100% одного ядра ЦПУ и 2 ГБ ОЗУ, то потребуется около 10 дней.

3. При частых запросах, например автоматически сгенерированных или с нескольких источников, скорость ответа на запросы начнёт падать.

Для пользователя, цель которого только производить списания со своих адресов или получать актуальные данные о балансе, вариант содержания полной сети менее привлекателен, чем использование стороннего решения. У клиентских программ, которые не загружают на устройство всю сеть, имеется несколько своих серверов с блокчейн сетью, напрямую недоступные пользователю. В таких случаях возникнут следующие проблемы:

1. Ограничение на частоту запросов. Из-за того что сервис предоставляет свои услуги нескольким пользователям им необходимо иметь большие ресурсы, чтобы обработать все запросы и в случае если клиент будет превышать допустимую квоту, то может получить временную блокировку на проведение запросов. Иначе безлимитные запросы будут предоставляется как отдельная платная услуга.

2. Приватность пользователя ставится под угрозу. Использование таких сервисов зачастую предполагает наличие аккаунта, а значит время авторизаций, IP адреса, запросы, криптовалютные адреса и другая активность клиента, будут фиксироваться на серверах компании. В дальнейшем данные могут быть перепроданы третьим лицам, если того не запрещает условия использования сервиса.

При использовании WEB клиентов недостатки локальных клиентов усиливаются и добавляются новые:

1. Отслеживание пользователя производится не только сервисом, но и счётчиками веб аналитики, такими как google analytics.

2. Зачастую пользователю, не предоставляются в распоряжение приватные ключи от адресов, либо сервис тоже имеет к ним доступ, а значит есть вероятность того, что средствами сможет распорядится не только их владелец.

Помимо этого для бизнес клиентам требуется значительно чаще совершать платёжные операции, а также контролировать большее число кошельков для получения средств или списания сразу с нескольких источников. Поэтому существующие системы будут не только неподходящими для таких условий взаимодействия, но и могут быть угрозой для успешного ведения дел.

2.3 Описание решения перечисленных проблем

Допустим, что пользователя не интересует вся структура блокчейн сети, т. е. ему не важна история транзакций других пользователей, размеры блоков и другая информация, обеспечивающая работу сети. Для пользователя

необходимо знать баланс средств на каких либо адресах и возможность сделать списание. Также ему необходимо максимально быстро получать этот баланс и производить запросы с большой частотой.

Решить проблемы с частотой запросов и сохранением приватности, можно только за счёт отказа от использования сторонних сервисов, т. е. мы должны хранить данные локально рядом с сервисом и напрямую взаимодействовать с блокчейн сетью.

Далее необходимо решить проблему с размером хранимой информации и скоростью синхронизации. Для уменьшения объёма хранимых данных нужно выделить только обязательные части блокчейн сети. Как упоминалось выше и изображалось на рисунке 4, для проведения платёжной операции необходимо знать идентификатор транзакции и номер выхода, который присваивается при её создании. Также учитывая, что пользователь хочет узнавать баланс именно адреса и находить не потраченные транзакции нужно будет по адресу. Перечень данных, необходимые для взаимодействия:

- Идентификатор транзакции
- Номер выхода в транзакции
- Адрес получения средств
- Количество полученных средств

В блокчейн сети не получится потратить одну транзакцию два раза, поэтому мы можем не хранить на клиенте всю цепочку транзакций, а записывать только новые созданные транзакции и удалять уже использованные, но это может создать ситуацию, когда необходимо будет узнать историю списаний по кошелькам и сделать это через текущий сервис будет невозможно.

Таким образом, чтобы устранить проблемы тонких клиентов, необходимо реализовывать толстый клиент, но сократив количество сохраняемых данных, получится уменьшить недостатки классических толстых клиентов, особенно если сохранять только информацию об интересующих кошельках.

2.4 Выбор инструментов и технологий

В первую очередь необходимо выбрать хранилище для данных, а для этого нужно знать с каким объёмом данных придётся работать и выбрать подходящую структуру хранения. В сценарии где мы получаем баланс пользователя, нам нужно по адресу получить весь список транзакций с количеством средств, в таком случае, нам нужно создать уникальный ключ, который приведёт к значению. Таким образом, мы можем отказаться от усложнённых реляционных баз данных и выбрать более простое, быстрое хранилище.

Размер данных. Согласно данным сервиса blockchain.com, количество не потраченных транзакций в системе Bitcoin, составляет около 73 миллионов на 5 марта 2019 года, график изображён на рисунке 5.

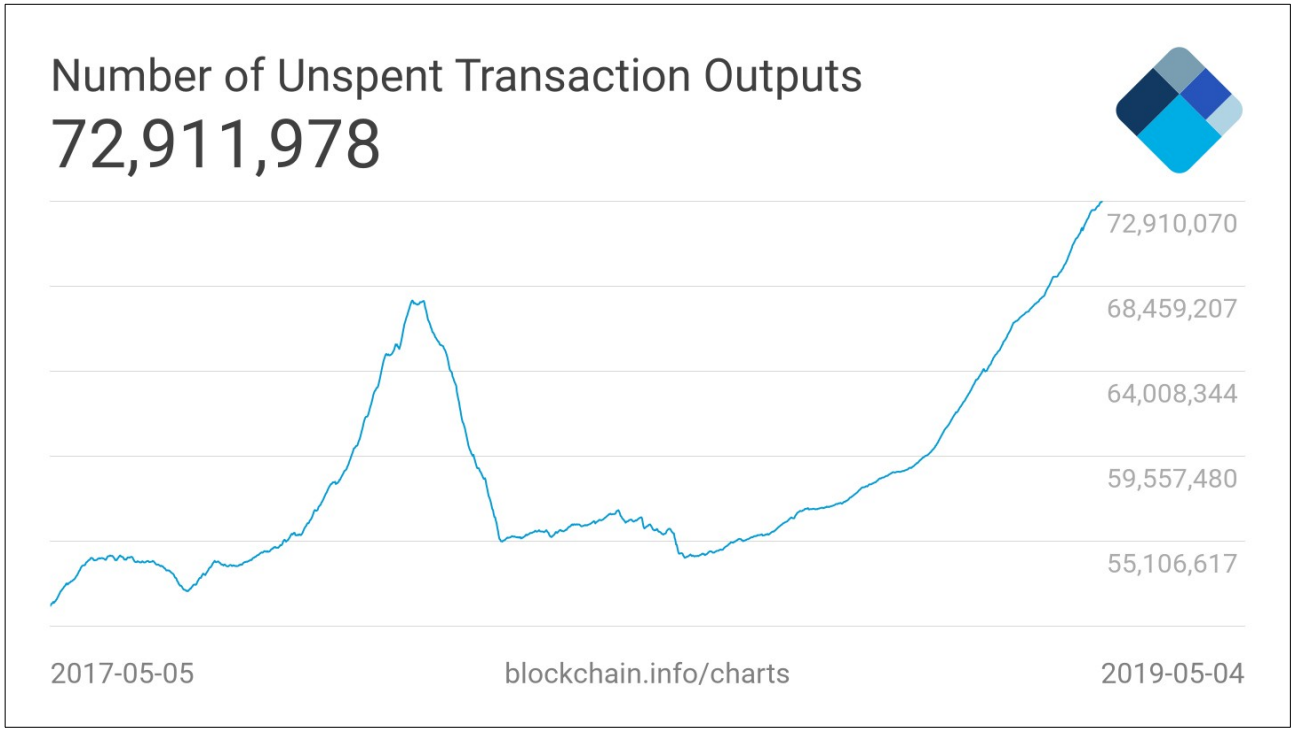

Рисунок 5. График количества не потраченных транзакций за 2 года.

Другими словами на сегодняшний момент, необходимо сохранить минимум 73 миллиона уникальных записей для всех адресов, учитывая, что мы будем хранить информацию только о своих адресах, то объём информации сокращается до несколько десятков тысяч записей за всё время работы сервиса.

Из-за простоты хранимых данных, можно выбрать базу данных со структурой хранения Ключ-Значение. Наиболее подходящим решением будет выбрать БД Redis, это сетевая, не реляционная база данных, с типом хранения ключ-значение. Данная система хранения данных давно зарекомендовала себя в корпоративной среде разработки, благодаря своим возможностям масштабирования в кластере и обеспечением надёжности хранения данных с включением встроенной опции репликации данных.

В качестве языка разработки отлично подойдёт язык Go, учитывая требования к многопоточной обработки запросов и обработке данных сети блокчейн. Помимо этого в стандартной библиотеке языка, реализована большая часть необходимых криптографических функций, а также благодаря сторонним разработчикам создано множество готовых вспомогательных функций для взаимодействия с блокчейн. Язык Go ускорит разработку HTTP REST API, за счёт созданных абстракций создателями языка и готовых решений сторонними разработчиками. Также данный язык программирования хорошо показывает себя в разработке приложений с микросервисной архитектурой, которая будет использована в реализации этого решения.

Для обеспечения своевременного получения обновлённых данных о кошельках другими сервисами и для уменьшения связности этих сервисов, можно воспользоваться система очередей сообщений. Использование данной системы позволяет добиться асинхронного взаимодействия. Среди множества решений Nats MQ является золотой серединой между системами с высокой производительностью и системами с удобным интерфейсом использования. Данная система будет использоваться для оповещения подключённых слушателей об изменениях балансов на интересующих их кошельках или целых групп кошельков.

2.5 Структура приложения и отдельные моменты реализации

В данном разделе представлена структура серверного приложения, состав компонентов, описаны отдельные моменты программной реализации и особенности взаимодействия между частями программы.

Данное приложение состоит из трёх основных компонентов:

1. HTTP REST API — компонент, реализующий пользовательское взаимодействие с приложением. Представляет собой HTTP запросы и заранее подготовленные форматы успешных ответов и ошибок.

2. Redis — абстракция для осуществления простого обращения в базу данных Redis из других компонентов приложения.

3. Chains — набор методов и абстракций, для скрытия особенностей работы с блокчейн сетями из HTTP компонента и сканера блокчейн сети.

4. Nats — абстракция над системой nats, позволяет взаимодействовать с очередью сообщений, в качестве источника данных.

5. Main — входная точка для запуска приложения, содержит в себе обработку конфигураций компонентов, их инициализация, запуск и бесконечное ожидание запросов HTTP компонента.

2.5.1 Особенности взаимодействия с блокчейн сетями

Проведя анализ существующих блокчейн сетей и криптовалют в первой главе проекта, можно сделать выводы, что взяв за основу сеть Bitcoin и реализовать взаимодействие с ней, то другие ответвления будут работать аналогично за исключением небольших деталей, ради которых и создавалось ответвление.

В модуле chains реализованы методы:

- для сканирование блокчейн сети, ожидание новых блоков и их обработка
- кэш данных для обработки перед записью в БД
- генератор новых транзакций

- вспомогательные функции, такие как конвертация приватного ключа в адрес, конвертация целочисленного представления валюты в число с плавающей точкой и т. д.

Некоторые криптовалюты такие как Ripple, позволяют подключится к их системе в качестве слушателя по протоколу Websocket и получать уведомления о новых блоках и транзакциях в момент их совершения, без периодичных запросов в блокчейн сеть.

Чтобы унифицировать процесс сканирования и проверку поступлениях новых данных по интересующим нас кошелькам, реализован объект checker, который запускает сканирование всех поддерживаемых криптовалют и инициирует добавление данных в БД.

Помимо этого необходимо реализовать надёжность хранения данных. Если по каким то причинам база данных станет недоступной или будут уничтожены все данные в ней, то необходим механизм восстановления кошельков. Для этого при запуске необходимо ввести mnemonic фразу, которая станет мастер ключом для всех остальных приватных ключей, на основе которых получаются адреса в блокчейн сетях. Поэтому если удалить все кошельки и оставить прежнюю мнемоник фразу, то процесс создания нового кошелька сгенерирует прежние адреса и приватные ключи. Далее можно будет повторно проверить сеть на поступления и расходы, чтобы восстановить все транзакции.

Список поддерживаемых криптовалют:

- BTC — Bitcoin
- BCH — Bitcoin Cash или BTCABC
- BTCSV — Форк от Bitcoin Cash
- DOGE — Dogecoin
- ETH — Ethereum
- XRP — Ripple

2.5.2 Особенности взаимодействия с базой данных

Важный момент при использовании базы данных типа Ключ-Значение это обеспечить уникальность сохраняемых ключей. В нашем случае мы будем генерировать уникальный ID для кошелька, который будет содержать в себе набор адресов и приватных ключей, для каждой поддерживаемой криптовалюты. Также можно использовать префиксы ключей, для удобной группировки в графическом режиме.

Примеры ключей в БД Redis:

- Отдельный кошелёк: *WALLETS:550e8400-e29b-41d4-a716-446655440000*
- Группа, содержит в себе список ID кошельков: *GROUPS:test*
- Транзакция на пополнение: *TxIn:WalletAddress:ABCD12312312AAAA*
- Транзакция на списание: *TxOut:WalletAddress:ABCD12312312AAAA*
- Не потраченная транзакция: *UTxO:WalletAddress:ABCD12312312AAAA*

Этих ключей достаточно чтобы реализовать систему, с возможностью поиска нужного адреса по кошелькам и расчёту его баланса. Все префиксы ключей и подключей Hash Map используемых в редисе содержатся в файле: */ChainService/redis/const.go*.

2.5.3 HTTP REST API для пользовательского взаимодействия

Чтобы у пользователь была возможность, управлять приложением, получать или отправлять информацию, были реализованы HTTP запросы, доступ к которым ограничен паролем в заголовке запроса. Ответы с данными содержаться в формате json.

Список реализованных запросов:

GET: /ping — отвечает кодом 200, служит для проверки доступности сервиса.

POST: /wallets?group=name — Создаёт новый кошелёк в группе name. Возвращает уникальный ID кошелька и адреса в различных валютах.

Пример ответа:


```
{
  "token": "0a2819e6-a37a-42c7-a2b0-607e52bd3fce",
  "btc": "1DwstFfz1X1oUhu638ou9rvZw52ASfeukw",
  "eth": "0x47bd8916ed80d9d4e5ccc89859c006d6b2063474",
  "xrp": "rDRXFSRRDvsAoaWig698ANZUkAu62LYnZj",
  "doge": "DJ5yRWcdJvv61i5gmioThd6ApCkTrTcEss",
  "btcabc": "qz8qyun8setsh8lzrymt96qw54qwk0n3yqw2w50tsd",
  "btcsv": "qz8qyun8setsh8lzrymt96qw54qwk0n3yqw2w50tsd"
}
```


GET: /wallets/{key}?val={token/group} — Возвращает баланс отдельного кошелька, по его токену или всей группы кошельков, необходимо передать в параметр key - слово token или group, а в параметр val - ID кошелька или название группы.

Пример ответа:


```
{
  "btc": 0.11101,
  "eth": 10.0,
  "xrp": 0,
  "doge": 0,
  "btcabc": 0,
  "btcsv": 0
}
```


POST: /wallets/token/0a2819e6-.../currency/btc?amount=0.1&addr=aaabbca

Запрос на списание средств с кошелька в определённой валюте (BTC).

POST: /wallets/group/name/currency/btc?amount=0.1&addr=aaabbca

Запрос на списание средств с группы кошельков в валюте (BTC).

В ответ приходит созданная транзакция в блокчейн сети, пример ответа:

{

Tx: abc1231231231231

}

GET: /wallets/token/open?val=0a2819e6-…

«Открывает» кошелёк по его ID, в ответ приходит список адресов.

Пример ответа:


```
{
  "token": "0a2819e6-a37a-42c7-a2b0-607e52bd3fce",
  "btc": "1DwstFfz1X1oUhu638ou9rvZw52ASfeukw",
  "eth": "0x47bd8916ed80d9d4e5ccc89859c006d6b2063474",
  "xrp": "rDRXFSRRDvsAoaWig698ANZUkAu62LYnZj",
  "doge": "DJ5yRWcdJvv61i5gmioThd6ApCkTrTcEss",
  "btcabc": "qz8qyun8setsh8lzrymt96qw54qwk0n3yqw2w50tsd",
  "btcsv": "qz8qyun8setsh8lzrymt96qw54qwk0n3yqw2w50tsd"
}
```


GET: /groups/wallet?val=0a2819e6 — Возвращает название группы кошелька.

Пример ответа:

{

"group": "name"

}

GET: /groups/ — Возвращает список существующих групп.

Пример ответа:

```
[
  "test_wallet"
]
```

GET: /wallets/ — Возвращает список существующих кошельков.

Пример ответа:

```
[
  "0a2819e6-a37a-42c7-a2b0-607e52bd3fce"
]
```

Компонент с HTTP API можно запустить в режим отладки, тогда в терминале запущенного приложения будет отображаться время выполнения каждого запроса. Это необходимо для временной оценки сложности определённых запросов в системе, для дальнейших оптимизаций системы.

2.5.4 Типы сообщений отправляемые в очередь сообщений

Чтобы другие сервисы могли своевременно получать информацию, о поступлениях или списаниях с их кошельков, была добавления система очереди сообщений Nats. Для того чтобы отдельный сервис получал только сообщения только по необходимым кошелькам, все отправляемые сообщения группируются по темам.

Есть две основные темы: TxIn и TxOut. Первая сообщает что появилась входящая транзакция на определённый кошелёк, вторая что было осуществлено списание. Эти две темы разбивается на подгруппы, которые состоят из названий групп кошельков, например: TxIn_GROUP_name и TxOut_GROUP_name. Они позволяют сервисам узнать, что в определённой группе кошельков есть поступление или списание.

Клиент очереди сообщений может подписаться как на поступление или списания всех групп кошельков или только на нужные ему группы, чтобы не тратить свои ресурсы на разбор чужих сообщений.

Сами сообщения о поступлениях или списаниях содержат разный набор полей.

Сообщение о поступлении содержит следующие поля:

- currency — валюта поступления (BTC, DOGE, ETH...)
- token — идентификатор кошелька
- group — группа кошелька
- hash — идентификатор транзакции
- address — адрес в блокчейн сети, на который поступили средства
- value — сумма поступления

Сообщение о списании содержит следующие поля:

- currency — валюта списания (BTC, DOGE, ETH...)
- token — идентификатор кошелька
- group — группа кошелька
- hash — идентификатор транзакции
- address_from — адрес в блокчейн сети, с которого списали средства
- address_to — адрес в блокчейн сети, на который поступили средства
- value — сумма списания

Сообщения отправляются, только после успешного сохранения в БД, чтобы в случае потери данных другие сервисы не получали ошибочную информацию о поступлениях или списаниях.

2.5.5 Схема взаимодействия системы и сторонних сервисов

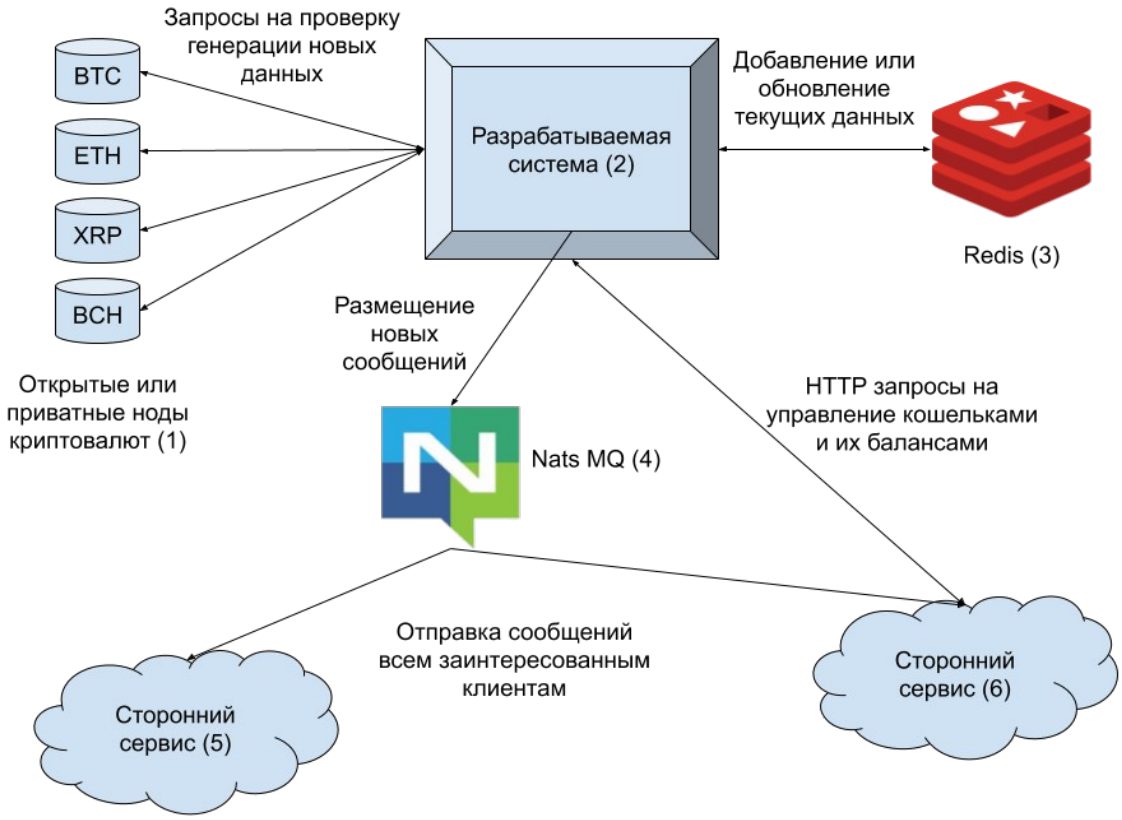

Рисунок 6. Схема взаимодействия системы и сторонних сервисов.

На рисунке 6 представлена схема взаимодействия системы с внутренними компонентами и внешними сторонними сервисами. Пояснения по пунктам представленных на рисунке:

1. Открытые или приватные ноды криптовалют — это точки доступа к полной информации о криптовалюте. Существуют открытые для общего пользования ноды и частные, закрытые. Система их использует для получения актуальной информации о блоках созданных в системе, а также для отправки собственных платёжных транзакций.

2. Разрабатываемая система содержит в себе всю логику по управлению цифровыми активами, предоставляет REST HTTP API для внешних сервисов.

3. Redis — база с типом хранения ключ-значение, используется как основное хранилище данных о кошельках, совершенных транзакций и балансах.

4. Nats MQ — система очереди сообщений, получает новые сообщения от системы, в качестве автора и отправляет всем подписанным в данный момент читателям.

5. Сторонний сервис, который взаимодействует с системой только в качестве слушателя обновлений по кошелькам. Такое может использоваться, когда сервис исполняет оповещает другие сервисы или запускает определённые процедуры по наступлению события.

6. Данный сервис пользуется системой в полном объёме, создаёт и группирует кошельки, производит списания и получает оповещения о поступление средств на свои адреса.

Разрабатываемая система берёт на себя всю сложность работы с различными криптовалютами, также периодично узнаёт об обновлениях в блокчейн сетях и моментально оповещает всех заинтересованных клиентов. Тем самым позволяет с лёгкостью автоматизировать операции по управлению криптовалютами, в масштабах предприятия.

2.6 Внедрение и эксплуатация полученного продукта

Процесс внедрения на предприятии без существующих систем контроля цифровых активов, можно разделить на следующие этапы:

1. Разработка технического задания и плана внедрения;
2. Конфигурирование программного продукта;
3. Запуск в тестовом окружении и тестирование системы;
4. Промышленная эксплуатация.

2.6.1 Разработка технического задания и плана внедрения

Необходимо удостоверится, что у заказчика имеются все необходимые ресурсы для нормального функционирования системы, в тестовом и промышленном окружениях.

Минимальные технические требования к серверному оборудованию:

- ОС на базе ядра Linux;
- Многоядерный процессор, минимум 2 доступных ядра;
- Оперативная память не менее 2 Гб;
- Доступное место в файловой системе от 10 Гб;
- Наличие в системе ПО docker, для запуска других необходимых систем;
- Доступ в сеть интернет и доступ к открытым или к частным криптовалютным нодам.

Для тестового запуска количественные характеристики можно сократить в 2 раза. Специальное ПО docker необходимо для запуска сопутствующих систем, таких как Nats и Redis. Данные системы необходимо запустить перед началом запуска основной системы, как его зависимости. Также стоит помнить, что в промышленном запуске, необходимо смонтировать в docker образ Redis файловую систему сервера, чтобы между перезапусками данные сохранялись.

2.6.2 Конфигурирование программного продукта

Конфигурирование системы происходит через конфигурационный JSON файл, путь до которого нужно указать при запуске системы.

Пример конфигурационного файла:

```
{
 "RestAPI": {
  "Debug": true,
  "Addr": "127.0.0.1:7101",
  "APIpwd": "REST_API_PASSWORD"
 },
 "Redis": {
  "Debug": true,
  "Addr": "127.0.0.1:6379",
  "DB": 0
 },
 "Chains": {
  "Debug": true,
  "Mnemonic": "mnemonic phrase",
  "NatsAddr": "127.0.0.1:4222",
  "Nodes": [{
   "Name": "BTC",
   "Enable": true,
   "Confirmations": 2,
   "URL": "http://111.111.111.111:8332",
   "Login": "login",
   "PWD": "password"
  }]
 }
```

Пояснения по некоторым полям:

- Debug — поле в объектах RestAPI, Redis и Chains позволяет включить вывод отладочной информации в данных модулях. Рекомендуется включать только в тестовом окружении.
- RestAPI.APIpwd — содержит текст, который необходимо указывать в заголовке «X-Token», при отправке HTTP запросов в систему.
- Redis.DB — указывает какую базу данных сервера Redis использовать для хранения данных. По умолчанию сервер Redis имеет 12 созданных баз данных, поэтому один сервер можно использовать разными сервисами.
- Chains.Mnemonic — Мнемоник фраза, которая является основой для генерации приватных ключей всех кошельков и адресов. Позволяет восстановить все созданные ключи если данные были утеряны.
- Chains.NatsAddr — Адрес по которому доступен Nats MQ сервер, обрабатывающий и пересылающий сообщения по нужным очередям.
- Chains.Nodes — содержит список точек доступа с полными данными о каждой криптовалюте. Точки доступа могут быть в открытом и закрытом доступе. Также есть поле Enable, которое позволяет отключить точку доступа, без удаления информации о ней из конфигурационного файла.

2.6.3 Запуск в тестовом окружении и тестирование системы

Для тестового запуска достаточно запустить сопутствующие системы в докер контейнерах с конфигурациями по умолчанию.

Для этого достаточно вызвать следующие команды:

```
docker run --name redis-test -p 6379:6379 -d redis
docker run --name nats-test -p 4222:4222 -d nats
```

Данные команды, автоматически загрузят с официального репозитория docker образов, готовые к запуску Redis и Nats сервера. Также автоматически их запустят и они будут доступны на localhost на портах указанных в ключе -p.

После тестирования можно остановить и удалить локальные образы этих контейнеров, следующими командами:

docker rm -f redis-test

docker rm -f nats-test

Контейнеры будут остановлены и удалены из локального хранилища.

Теперь можно запустить скомпилированную систему с тестовым конфигурационным файлом, следующей командой:

./ChainService -conf ./test_config.json

При невозможности соединится с сервером Redis или Nats, программа сообщит об этой ошибке и прекратит работу. Сигналом о том что система запущена и готова к работе, служит сообщение в логах:

[HTTP] Started on 127.0.0.1:7101

Это сообщение означает что система проинициализировала все подключения к другим серверам, смогла запустить собственный HTTP сервер на указанном порту и готова обрабатывать входящие сообщения.

Для проверки работоспособности, можно вызвать несколько HTTP запросов представленных в разделе 2.5.3. Также слудует провести комплексное тестирование, зачисления и списания средств на тестовых точках доступа криптовалют.

2.6.4 Промышленная эксплуатация

Процесс запуска в промышленном окружение идентичен процессу запуска в тестовых условиях, за исключением некоторых моментов.

Образ Redis контейнера, необходимо запускать с дополнительным ключом - v, в котором необходимо указать путь до директории в текущем сервере, для хранения данных вне контейнера.

docker run -v redis-data:./data/redis --name redis-prod -p 6379:6379 -d redis

Данная команда запустит контейнер с Redis сервером, который будет использовать указанное хранилище сервера, вместо хранилища внутри контейнера. Это необходимо сделать, т. к. docker контейнеры после полной остановки очищают все изменения внутри контейнера, созданные за время своей работы. Без учёта типа данных, служебные, системные или пользовательские. удаления контейнеров аналогичные.

Запустить целевую систему с конфигурационным файлом для промышленной работы можно, следующей командой:

./ChainService -conf ./prod_config.json

В prod_config.json необходимо убедится, что мнемоник фраза отличается от фразы в тестовой файле. Иначе на этих окружениях будут генерироваться одинаковые ключи и адреса кошельков.

Для автоматического перезапуска или упрощённого развёртывания системы, можно поместить в docker контейнер саму систему и использовать docker-compose систему, которая будет запускать все эти сервисы и следить за их доступностью. В случае проблем с сервисами данная система их автоматически перезапускает.

Выводы по второй главе

Во второй главе был описан выбор инструментов, процесс разработки программного сервиса и решения проблем возникшие в ходе работы. Получившееся программное обеспечение удалось успешно запустить и создать несколько кошельков в различных криптовалютах. Сторонние сервисы получали все уведомления об изменениях через очередь сообщений и могли отправлять HTTP запросы, для управления своими кошельками и средствами на них.

Для упрощения взаимодействия с блокчейн сетью, были использованные библиотеки сторонних разработчиков такие как:

- github.com/btcsuite/btcd
- github.com/btcsuite/btcutil
- github.com/cpacia/bchutil
- github.com/piotrnar/gocoin

Для работы с базой данных библиотека: github.com/go-redis/redis.

Для создания HTTP REST API, использовался маршрутизатор HTTP запросов из пакета github.com/go-chi/chi.

ЗАКЛЮЧЕНИЕ

В ходе работы над данным курсовым проектом, были рассмотрены и изучены одни из самых популярных платформ, которые позволяют проводить различные операции с современными цифровыми активами. Проведён анализ текущих реализаций, для выявления недостатков их реализаций, чтобы создать максимально полезный и удобный сервис для удовлетворения потребностей бизнес пользователей. Изучение особенностей работы технологии блокчейн позволило в кратчайшие сроки создать продукт, который будет взаимодействовать с различными криптовалютами, не прибегая к дублированию кода и его функционала.

Во время разработки программного продукта был использован язык программирования Go и библиотеки сторонних разработчиков, для упрощения взаимодействия с блокчейн. В качестве базы данных была использована Redis от компании Redis Labs, позволяющая хранить данные в виде Ключ-Значение.

Результатом данной работы является готовый к распространению и эксплуатации программный код, компилируемый под Windows и Linux операционные системы. Продукт в дальнейшем можно дополнять другими криптовалютами как и Bitcoin подобными, так и совсем из других ответвлений. Также данный продукт был введён в эксплуатацию на ресурсах одного из клиентов и успешно используется в настоящее время.

К полученным выводам можно отнести, подтверждение мысли о том, что готовые и массово используемое программные продукты, могут содержать недостатки, которые мешают использовать их в условиях повышенной частоты

запросов или контроля по различным признакам. Из этого можно сделать вывод, что всегда есть место для улучшения или создания совершенно нового продукта, со своей уникальной структурой.

При стремительном развитии информационных технологий, буквально ежедневно появляются новые способы взаимодействия клиентов и организаций. Специалисту, который хочет остаться востребованным на рынке, необходимо изучать и стараться использовать в профессиональной деятельности, чтобы сложить всю картину и понять почему нынешние решения именно такие и почему они более популярны у пользователей.

Проведённая работа с теоретическим материалом, а также практическая деятельность, позволяет сказать, что, поставленные задачи выполнены и цель данного проекта достигнута.

СПИСОК ИСТОЧНИКОВ

1. Аксенов Д.А. — Направления и особенности применения блокчейн-технологии в экономике / Аксенов Денис Алексндрович, Куприков Антон Петрович, Саакян Пайлак Андроникович — Спб.: «Научно-технические ведомости СПбГПУ. Экономические науки.», 2018. — 30с.

2. Бентли Д. Жемчужины программирования 2 изд. / Джон Бентли — Спб: «Питер», 2013. — 272с.

3. Донован А.А. Язык программирования Go / Донован Алан А. А. — М.: «Вильямс», 2016. – 432с.

4. Макконнелл С. Совершенный код. / Стив Макконнелл — М: «Русская редакция», 2017. — 896с.

5. Макконнелл С. Профессиональная разработка программного обеспечивания. / Стив Макконнелл — Спб.: «Символ-плюс», 2016. — 240с.

6. Мартин Р. Чистый код: Создание, анализ, рефакторинг. / Роберт Мартин — Спб.: «Питер», 2015. — 464с.

7. Мартин Ф. Рефакторинг. Улучшение существующего кода. / Фаулер Мартин — Спб.: «Символ-плюс», 2017. — 432с.

8. Орам Э. — Идеальный код. / Энди Орам — Спб.: «Питер», 2014. - 624с.

9. Равал С. Децентрализованные приложения. Технология Blockchain в действии / Сирадж Равал — Спб.: «Питер», 2017. — 378с.

10. Саммерфильд М. Программирование на Go Разработка приложений XXI века. / Марк Саммерфильд — М.: ДМК Пресс, 2016. – 580с.

11. Тапскотт А. — Технология блокчейн - то, что движет финансовой революцией сегодня. / Алекс Тапскотт — М.: «Эксмо», 2017. – 483с.

12. Цилюрик О. — QNX/UNIX: анатомия параллелизма. / Олег Цилюрик — Спб.: «Символ-плюс», 2016. — 288с.

13. Dr. Shermin Voshmgir. Blockchains & Distributed Ledger Technologies // Интернет-ресурс для обучения применения блокчейн технологий на практике

— 2018. — 19 января [Электронный ресурс]. URL: https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/

14. Google Inc. The Go Programming Language // Интернет-ресурс обучения языку программирования Go со встроенным sandbox компилятором — 2019 — 3 марта [Электронный ресурс]. URL: https://golang.org/

15. Jerry Brito, Andrea Castillo. Bitcoin: A Primer for Policymakers // Научная статья университета Джорджа Мейсона — 2016. — 26 января [Электронный ресурс]. URL: https://www.mercatus.org/system/files/Brito_BitcoinPrimer.pdf

16. Jimmy Aki. Leading Blockchain and Gaming Companies Form Blockchain Game Alliance // Научно популярный интернет журнал о цифровых активах — 2018. — 27 марта [Электронный ресурс]. URL: https://bitcoinmagazine.com/articles/leading-blockchain-and-gaming-companies-form-blockchain-game-alliance/

17. Nathan Popper. A Venture Fund With Plenty of Virtual Capital, but No Capitalist // Интернет издание газеты The New York Times — 2016. — 21 мая [Электронный ресурс]. URL: https://www.nytimes.com/2016/05/22/business/dealbook/crypto-ether-bitcoin-currency.html

18. Альтернативная реализация полного узла сети на Go // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/btcsuite/btcd

19. Библиотека для взаимодействия с Redis базой на Go // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/go-redis/redis

20. Готовые методы и типы для сети Bitcoin // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/btcsuite/btcutil

21. Готовые методы и типы для сети Bitcoin Cash // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/cpacia/bchutil

22. Гуляев Р.А. Криптовалюты: сущность, эволюция и становление в качестве средства платежа // Информационный ресурс электронных научных работ — 2018. — 20 марта [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/kriptovalyuty-suschnost-evolyutsiya-i-stanovlenie-v-kachestve-sredstva-platezha

23. Демидов А. Proof-of-Work описание механизма // Информационный ресурс об информационных финансовых технологиях — 2018. 24 июля [Электронный ресурс]. URL: https://coinspot.io/beginners/proof-of-work-dlya-chajnikov/

24. ИА Банки.ру // Информационный ресурс — 2019. — 26 декабря [Электронный ресурс]. URL: https://www.banki.ru/wikibank/tsifrovoy_aktiv_/

25. ИА Банкир.Ру // Информационный ресурс — 2016. — 19 апреля [Электронный ресурс]. URL: https://bankir.ru/events/9-forum-blokchein-i-otkrytye-platformy-2016/

26. Н. А. Олифер. Вызов удаленных процедур (RPC) // Информационный ресурс Центра ИТ МГУ им. М.В. Ломоносова — 2016. — 25 декабря [Электронный ресурс]. URL: http://docstore.mik.ua/sos/glava_12.htm#_3_2

27. Описание Web сервиса Blockchain Wallet // Официальный сайт проекта blockchain.com — 2020. — 11 января [Электронный ресурс]. URL: https://www.blockchain.com/en/wallet

28. Описание Web сервиса Coinbase // Официальный сайт проекта Coinbase — 2020. — 11 января [Электронный ресурс]. URL: https://www.coinbase.com/

29. Описание платформы Waves // Официальный сайт проекта Waves — 2020. — 26 января [Электронный ресурс]. URL: https://wavesplatform.com/technology

30. Описание платформы Ethereum // Официальный сайт проекта Ethereum — 2019. — 10 декабря [Электронный ресурс]. URL: https://ethereum.org/what-is-ethereum/

31. Описание ПО Bitcoin Core // Официальный сайт проекта Bitcoin — 2020. — 10 января [Электронный ресурс]. URL: https://bitcoin.org/ru/wallets/desktop/linux/bitcoincore/

32. Описание ПО Ethereum Wallet // Официальный сайт проекта Ethereum Wallet — 2020. — 10 января [Электронный ресурс]. URL: https://www.myetherwallet.com

33. Описание ПО Exodus // Официальный сайт проекта Exodus — 2020. — 11 января [Электронный ресурс]. URL: https://www.exodus.io/desktop

34. Описание ПО Jaxx // Официальный сайт проекта Jaxx — 2020. — 11 января [Электронный ресурс]. URL: https://jaxx.io/

35. Описание проекта Monero // Официальный сайт проекта Monero — 2019. — 18 октября [Электронный ресурс]. URL: https://web.getmonero.org/technical-specs/

36. Описание проекта Ripple // Официальный сайт проекта Ripple — 2019. — 13 ноября [Электронный ресурс]. URL: https://ripple.com/xrp/

37. Описание проекта Zcash // Официальный сайт проекта Zcash — 2019. — 25 сентября [Электронный ресурс]. URL: https://z.cash/ru/technology/

38. Полное решения для работы с Bitcoin сетью на Go // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/piotrnar/gocoin

39. Реализация маршрутизатора HTTP на Go // Web хостинг репозиториев открытого исходного кода — 2019. — 15 марта [Электронный ресурс]. URL: https://github.com/go-chi/chi

40. Ричард Троманс. Hype Killer – Only 1% of Companies Are Using Blockchain // Интернет журнал о юриспруденции в цифровом бизнесе — 2018. — 4 мая [Электронный ресурс]. URL:

https://www.artificiallawyer.com/2018/05/04/hype-killer-only-1-of-companies-are-using-blockchain-gartner-reports/

41. Семенов Ю.А. Протоколы Интернет // Информационный ресурс ИТЭФ-МФТИ — 2015. — 13 января [Электронный ресурс]. URL: http://book.itep.ru/4/44/inter_44.htm

42. Филдинг Т.Р. Архитектурный стиль и дизайн сетевого ПО. Архитектура REST // Информационный ресурс Университета Калифорнии в Ирвайне — 2015. — 16 Марта [Электронный ресурс]. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

ПРИЛОЖЕНИЕ 1

Исходный код программы данного курсового проекта, помещён в несжатый zip архив «01_Исходный_код.zip» и записан на компакт диск.

Список файлов на диске:

- 01_Исходный_код.zip
- 02_Презентация_проекта.pdf