Санкт-Петербургский государственный университет

Программная инженерия
Кафедра системного программирования

Погребной Дмитрий Андреевич

Разработка платежной системы для взаимодействия с различными реализациями blockchain

Курсовая работа

Научный руководитель:
ст. преп. Я. А. Кириленко

Технический консультант:
Lead developer DSX Technologies Ф. П. Долголев

Санкт-Петербург
2020

Оглавление

Введение

В современном мире существует множество видов цифровых валют. Один из них — криптовалюты, в основе которых лежат децентрализованная система и принципы криптографии. В 2008 году была опубликована оригинальная статья [8] о реализации первой криптовалюты Bitcoin. В этой же статье описывается технология blockchain — распределенная децентрализованная база данных, которая обеспечивает неподдельность истории всех операций и допускает только строго консистентное изменение данных. На момент мая 2020 года мировой рынок насчитывает более 4500 разных криптовалют [2], многие из которых базируются на технологии blockchain.

Отличительные особенности и постоянное развитие инфраструктуры криптовалют делают их более привлекательными для пользователей. Появляется множество разнообразных организаций связывающих свою деятельность с криптовалютами.

В связи с ростом популярности данного направления и множеством различных реализаций blockchain, возникает потребность в электронной платежной системе, которая скроет детали реализации работы с различными криптовалютами и предоставит унифицированный интерфейс для выставления счетов, приема и отправки платежей. Такая система позволит торговым компаниям и индивидуальным предпринимателям использовать криптовалюты как платежное средство для совершения финансовых операций.

1. Постановка задачи

Целью данной курсовой работы является разработка платежной системы, позволяющей взаимодействовать с различными криптовалютами на основе blockchain и производить стандартные операции участникам финансовых отношений.

Для достижения цели были поставлены следующие задачи:

- Сделать обзор предметной области
- Провести анализ существующих решений
- Доработать прототип платежной системы
- Реализовать поддержку криптовалюты Gram

2. Обзор предметной области

2.1. Blockchain

Технология blockchain основана на использовании распределенной базы данных, которая представляет собой список блоков. Каждый блок состоит из набора транзакций, идентификатора предыдущего блока и набора некоторых других данных. Идентификатор — это хеш от содержащихся в блоке данных. Таким образом, изменение данных в любом блоке, который уже принят сетью, изменяет его идентификатор, что делает всю остальную цепочку невалидной. Такая связь блоков делает невозможным изменение истории транзакций. Этот базовый принцип blockchain имеет множество расширений и модификаций, которые повышают эффективность использования blockchain для конкретных задач.

Помимо криптовалют технология blockchain применяется в системах электронного голосования, интернете вещей, энергетике, области распространения контента и многих других сферах [1].

Первые поколения blockchain для криптовалют [3] имели низкую пропускную способность, что ограничивало их масштабируемость. В современных реализациях технологии blockchain пропускная способность многократно увеличилась.

2.1.1. Консенсус

Децентрализация системы обеспечивается отсутствием доверенного регулятора, контролирующего процесс валидации транзакций, и применением протокола консенсуса. Протокол консенсуса в blockchain — это набор правил, по которым узлы распределенной сети blockchain достигают консенсуса при валидации транзакций и блоков. Майнеры — это участники сети, которые заинтересованы в получении вознаграждения за создание новых блоков в сети blockchain. Существует много различных протоколов консенсуса [10]. Наиболее популярные из них Proof-of-Work, Proof-of-Stack и Proof-of-Burn.

Идея протокола Proof-of-Work состоит в том, что майнеру для формирования нового блока необходимо выполнить трудную вычислительную задачу, результат которой можно быстро проверить. Чтобы новый блок был признан валидным необходимо, чтобы результат такой задачи оказался корректен. Таким образом майнер блока доказывает, что была проделана работа по решению задачи и новый блок является валидным. Данный протокол консенсуса требует большой вычислительной мощности сети для обеспечения высокой устойчивости к различным атакам.

Протокол Proof-of-Stack основан на том, что майнеру для формирования нового блока необходимо на время резервировать часть своих средств для возможности создания новых блоков. Вероятность формирования майнером очередного блока в blockchain пропорциональна доле, которую составляет количество криптовалюты, зарезервированной этим участником сети, от общего количества зарезервированных средств. Этот протокол не требует большой вычислительной мощности сети, однако мотивирует к накапливанию средств участниками сети, что потенциально может приводить к централизации сети.

Механизм протокола Proof-of-Burn заключается в том, что майнеру для формирования нового блока необходимо отправлять криптовалюту на специальный счет, с которого нельзя вернуть или потратить валюту. Таким образом участник сети сжигает часть своей криптовалюты, получая возможность создавать новые блоки. Вероятность формирования майнером очередного блока пропорциональна доле, которую составляет количество сожженной криптовалюты участника сети за все время от общего количества сожженных средств. Аналогично протоколу Proof-of-Stack данный протокол не требует больших вычислительных мощностей и потенциально может приводить к централизации сети. Протокол Proof-of-Burn мотивирует участников сети на долгосрочное участие в проекте, что повышает стабильность курса криптовалюты, в основе которой лежит данный механизм.

Особенности используемого протокола и конкретного алгоритма консенсуса во многом определяют такие характеристики blockchain как устойчивость к различного рода атакам, скорость транзакций, правила

валидации транзакций и блоков, требования к оборудованию и ресурсам для содержания узла сети.

2.2. Fork-производные криптовалюты

Fork-производная криптовалюта появляется при hard fork blockchain существующей криптовалюты. Hard fork blockchain — это изменение в программном обеспечении узлов сети blockchain, при котором новые блоки, добываемые на основе нового программного обеспечения, не считаются действительными для узлов сети, работающих на старом программном обеспечении. Такое изменение в программном обеспечении узлов приводит к возникновению двух валидных цепочек блоков: одна цепочка является валидной для узлов сети со старым программным обеспечением, а другая для узлов с новым программным обеспечением. Таким образом исходный blockchain расщепляется на два blockchain: старый и новый.

Наиболее популярные fork-производные криптовалюты [2]: Litecoin (LTC), Bitcoin Cash (BTH), Bitcoin Gold (BTG) fork-производные от Bitcoin (BTC) и Ethereum (ETH) fork-производная от Ethereum Classic (ETC).

2.3. Электронные платежные системы

Электронная платежная система - это разновидность системы расчётов между финансовыми субъектами, которая проводит все транзакции через Интернет. Для обеспечения безопасности финансовых операций в сети используются различные протоколы, наиболее популярный из них 3-D Secure [11].

В электронных платежных системах существуют процедуры возврата средств. Возвратный платеж (chargeback) — это процесс оспаривания плательщиком операции оплаты товара или услуги, произведенной получателем денежных средств. После возврата средств плательщику, доказательство истинности операции возлагается на получателя. Chargeback выполняется при отмене двойного списания, возврате

средств за некачественный товар или услугу а также по другим причинам.

Банковские электронные платежные системы ведут свою деятельность по принципу "Знай своего клиента". Этот принцип обязывает финансовые институты идентифицировать личность контрагента перед проведением финансовой операции. Данный принцип направлен на мониторинг финансовых операций и предотвращение коррупции и взяточничества. Следование принципу "Знай своего клиента" обеспечивается соответствующими законами.

Помимо финансового процессинга, современные платежные системы предоставляют широкий спектр услуг, а также программные модули для интеграции в различные системы управления контентом (CMS). За каждую финансовую операцию платежная система взимает комиссию, которая зависит от конкретной системы и суммы платежа.

3. Обзор существующих решений

На данный момент существуют коммерческие сервисы, которые предоставляют криптовалютные шлюзы для совершения финансовых операций. Такие услуги полезны для тех компаний, которые хотят предоставить своим клиентам возможность оплаты услуг современным видом платежных средств, но не имеют достаточно ресурсов для освоения передовых технологий в сфере финансов.

После анализа платежных систем были выделены следующие параметры для сравнения:

- Количество поддерживаемых криптовалют
- Способ предоставления услуг
- Способ хранения средств
- Открытость исходного кода

Количество поддерживаемых полноценных криптовалют непосредственно определяет возможности конкретной платежной системы.

Способ предоставления услуги влияет на необходимость хостинга платежной системы на стороне пользователя. Обычно используются две модели предоставления услуги: SaaS и Self-hosted. Предоставление программного обеспечения как услуги (SaaS) снимает с пользователей необходимость хостинга, что влечет за собой взятие абонентской платы. Модель Self-hosted обычно не подразумевает взимание абонентской платы, вместо этого пользователю необходимо позаботиться о сервере, на котором будет развернута платежная система.

По способу хранения средств платежные системы подразделяются на кастодиальные и некастодиальные. Кастодиальные сервисы хранят средства пользователей на своих кошельках и предоставляют возможность вывода средств, взимая при этом некоторую комиссию. Такие сервисы также берут на себя все аспекты, касающиеся безопасности хранения средств на счетах. Некастодиальные сервисы переводят средства пользователей напрямую, что обычно уменьшает задержку перевода и

комиссию. Пользователи таких сервисов сами занимаются безопасностью хранения своих средств.

Открытый исходный код платежной системы позволяет рассмотреть детали реализации архитектуры и убедиться в надежности сервиса. Однако подобное программное обеспечение почти всегда является проприетарным, а значит доступ к исходному коду подобных систем отсутствует.

В таблице 1 представлено сравнение некоторых популярных сервисов для финансовых операций с криптовалютами.

Сервис	Валюты	Хостинг	Кастодиальный	Open source
BestRate[1]	>120	SaaS	+	-
B2BinPay[2]	19	SaaS	+	-
AliantPayments[3]	3	SaaS	-	-
CoinGate[4]	46	SaaS	+	-
PayCoiner[5]	18	SaaS	+	-
CoinPayments[6]	92	SaaS	+	-
BitPay[7]	5	SaaS	+	-
BTCPay Server[8]	12	Self-hosted	-	+
Cashier-BTC[9]	BTC	Self-hosted	-	+

Таблица 1: Популярные криптовалютные платежные системы

Сравнение показало, что большинство платежных сервисов являются кастодиальными, предоставляются по модели SaaS и имеют закрытый исходный код. Продукты BtcPay Server и Cashier-BTC являются некастодиальными self-hosted проектами с открытым исходным кодом. Однако эти системы работают только с небольшим подмножеством популярных криптовалют. Первый сервис поддерживает работу с Bitcoin и его fork-производными, а второй только с Bitcoin.

[1]https://bestrate.org/

[2]https://b2binpay.com/

[3]https://aliantpayments.com/crypto-processing/

[4]https://coingate.com/

[5]https://paycoiner.com/

[6]https://coinpayments.net/

[7]https://bitpay.com/

[8]https://btcpayserver.org/

[9]https://github.com/Overtorment/Cashier-BTC

4. Разработка платежной системы

Разработка платежной системы Blockchain payment system[10] (BPS) проводилась на базе прототипа платежной системы, который унифицирует взаимодействие с различными blockchain криптовалют. Этот прототип является результатом [13] курсовой работы Сергея Скаредова 2019 года. Прототип написан на языке Kotlin и поддерживает три криптовалюты: Bitcoin, Ripple, TRON.

4.1. Архитектура прототипа платежной системы

Система состоит из двух модулей. Первый из них, Core, представляет собой абстракцию, которая предоставляет пользователю необходимые методы для оперирования валютой. Второй модуль, Crypto, отвечает за непосредственную работу с различными криптовалютами.

4.1.1. Модуль Core

Модуль Core содержит три компоненты и оперирует сущностями платеж (Payment) и счёт (Invoice). Компонента PaymentProcessor отвечает за создание объектов Payment и обновление их статуса.
InvoiceProcessor отвечает за создание объектов Invoice и мониторинг их статуса. Manager координирует работу всей системы.

4.1.2. Модуль Crypto

Модуль Crypto содержит три компоненты. CoinClient описывает интерфейс, с помощью которого модуль Core взаимодействует с blockchain. Компонента BlockchainListener объявляет методы мониторинга и публикации новых транзакций, которые нужно проверить InvoiceProcessor. NodeConnector реализует функциональность по установлению соединения с узлом сети и отправке запросов.

Исходная архитектура системы представлена на рис. 1.

[10]https://github.com/dsx-tech/Blockchain-Payment-System

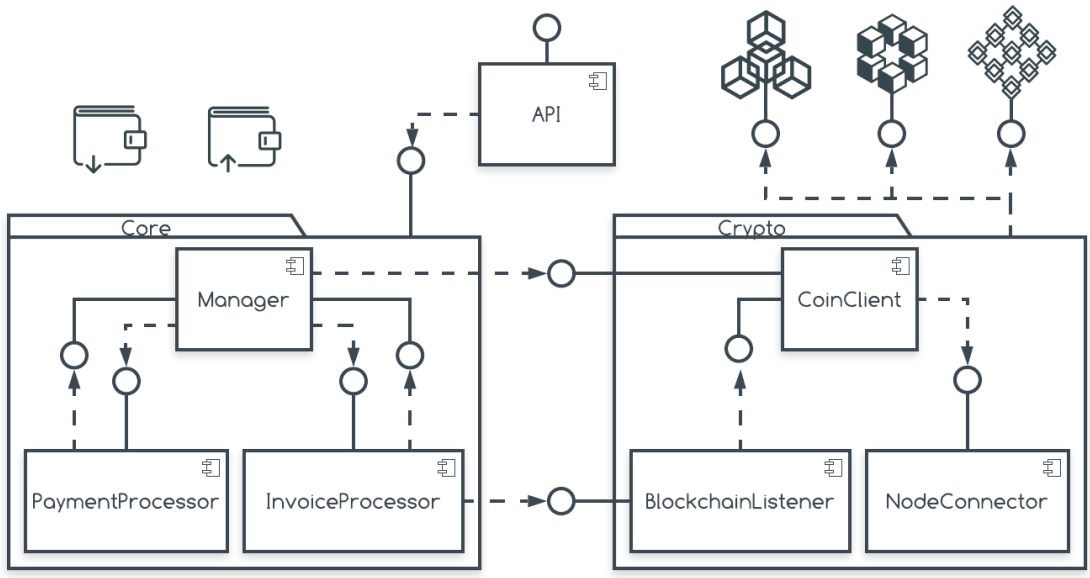

Рис. 1: Начальная архитектура BPS

Более подробное описание архитектуры прототипа содержится в статье Сергея Скаредова [13].

4.2. Доработка прототипа платежной системы

В результате проведенной работы были сделаны некоторые улучшения архитектуры. Появился новый компонент CoinsManager, который содержит всю логику работы с CoinClient и предоставляет интерфейс управляющему компоненту BPSManager, а он в свою очередь отвечает за управление всей системой в целом.

Для работы с каждой криптовалютой требуется множество параметров: приватный ключ, адрес, параметры для подключения к узлу сети и другие. Раньше все эти значения были прописаны напрямую в исходном коде системы. Для того чтобы исправить это, добавлена поддержка типобезопасного конфигурирования системы с помощью библиотеки konf[11]. Библиотека konf выбрана не случайно. Её аналог, библиотека Hoplite[12], обладает схожей функциональностью, но реализует другой принцип встраивания в систему и труднее интегрируется в те-

[11]https://github.com/uchuhimo/konf
[12]https://github.com/sksamuel/hoplite

кущий проект. Именно поэтому выбор остановился на konf. Также добавился компонент Config, который содержит структуры данных для конфигурации каждой поддерживаемой криптовалюты.

Прототип платежной системы представляет собой обычную библиотеку. Полноценная платежная система должна иметь возможность запускаться как отдельное приложение и предоставлять API для взаимодействие с ней по сети. Для того, чтобы BPS стала полноценной системой, реализовано REST API с помощью фреймворка ktor[13]. ktor является наиболее популярным и активно развивающимся фреймворком для создания сетевых приложений на Kotlin, а также имеет исчерпывающую документацию. Реализованная функциональность соответствует компоненте REST API на диаграмме компонентов платежной системы.

Помимо улучшения прототипа системы, в рамках изучения кодовой базы и проверки ее на корректность, были написаны unit-тесты для всей существующей функциональности системы. Для написания unit-тестов использовались две Java библиотеки JUnit5[14] и Mockito[15]. Использование Java библиотек не влечет за собой никаких сложностей, так как Kotlin полностью совместим с Java. JUnit5 де факто является стандартной библиотекой для написания unit-тестов на языке Java, а Mockito выбрана из-за своей выразительности и полной совместимости с JUnit. Благодаря unit-тестам удалось исправить несколько мелких ошибок в исходном коде.

Архитектура после перечисленных изменений представлена на рис. 2.

Модуль Database, отвечающий за работу платежной системы с базой данных, был реализован Артемом Луневым в рамках его курсовой работы 2020 года.

[13]https://github.com/ktorio/ktor
[14]https://github.com/junit-team/junit5
[15]https://github.com/mockito/mockito

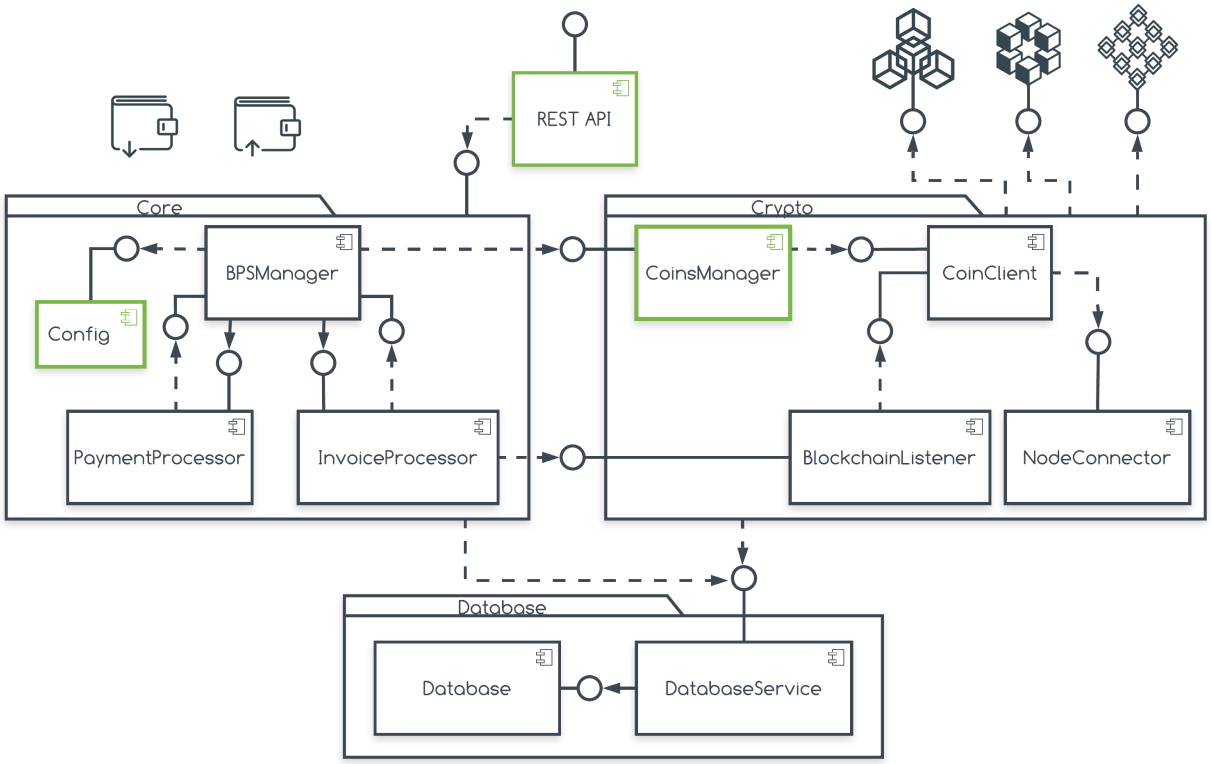

Рис. 2: Доработанная архитектура BPS

5. Поддержка криптовалюты Gram

Telegram Open Network (TON) — это платформа, построенная на принципе оверлейной P2P-сети [9], имеющая сервисы для обмена сообщениями, платежных операций в криптовалюте Gram (GRM), сервисы для хранения данных, а также сервис для поддержки распределенных приложений. TON Blockchain — blockchain пятого поколения [3], который является ядром сервиса платежных операций в криптовалюте Gram. Его отличительной особенностью является высокая скорость транзакций, которая во много раз превышает показатели других blockchain. А также уникальный протокол консенсуса, который дает возможность исправления ошибок в ближайшей истории операций, что потенциально снижает вероятность образования его fork'ов.

TON написан на языке C++14 [6]. Исходный код платформы находится в открытом доступе[16]. Документация TON представляет собой

[16]https://github.com/ton-blockchain/ton

white paper [3] и несколько других материалов[17], которые идейно описывают принцип работы компонентов платформы. Подробная документация TON, к сожалению, отсутствует, поэтому при реализации поддержки криптовалюты Gram в BPS приходилось разбираться в недокументированном исходном коде и заниматься реверс-инжинирингом отдельных частей системы.

Платформа TON разрабатывалась как open-source проект командой Telegram с 2018 года. 12 мая 2020 года основатель мессенджера Telegram, Павел Дуров, публично объявил о прекращении разработки и поддержки платформы TON командой Telegram[18]. На этот момент уже была запущена тестовая сеть, TON находился на стадии тестирования.

Прекращение поддержки TON командой Telegram не означает, что проект полностью свернут и не будет запущен. Команда разработчиков компании TON Labs[19], которая также участвовала в разработке TON, работает над самостоятельным запуском платформы, но уже под названием Free TON. Уже запущена тестовая сеть платформы. Помимо команды TON Labs вокруг платформы TON сформировалось активное сообщество независимых разработчиков, которое также заинтересовано в запуске платформы. Дальнейшую судьбу платформы Free TON предсказать не представляется возможным.

Исходный код Free TON является open-source. Некоторая часть исходного кода Free TON написана на Rust[20], весь остальной код взят из исходного кода платформы TON. Принципы работы Free TON полностью совпадают с TON. Логика работы с криптовалютой Gram и взаимодействие с узлами сети платформы в Free TON и TON не имеют отличий. Таким образом, реализованная поддержка криптовалюты Gram в TON корректна и для платформы Free TON.

Криптовалюта Gram выбрана для поддержки из-за перспективности платформы TON, высокой пропускной способности TON Blockchain и

[17]https://test.ton.org

[18]https://telegra.ph/What-Was-TON-And-Why-It-Is-Over-05-12

[19]https://tonlabs.io

[20]https://github.com/tonlabs

отсутствия поддержки данной криптовалюты в других платежных системах.

Архитектура модуля поддержки криптовалюты Gram представлена на рисунке рис. 3.

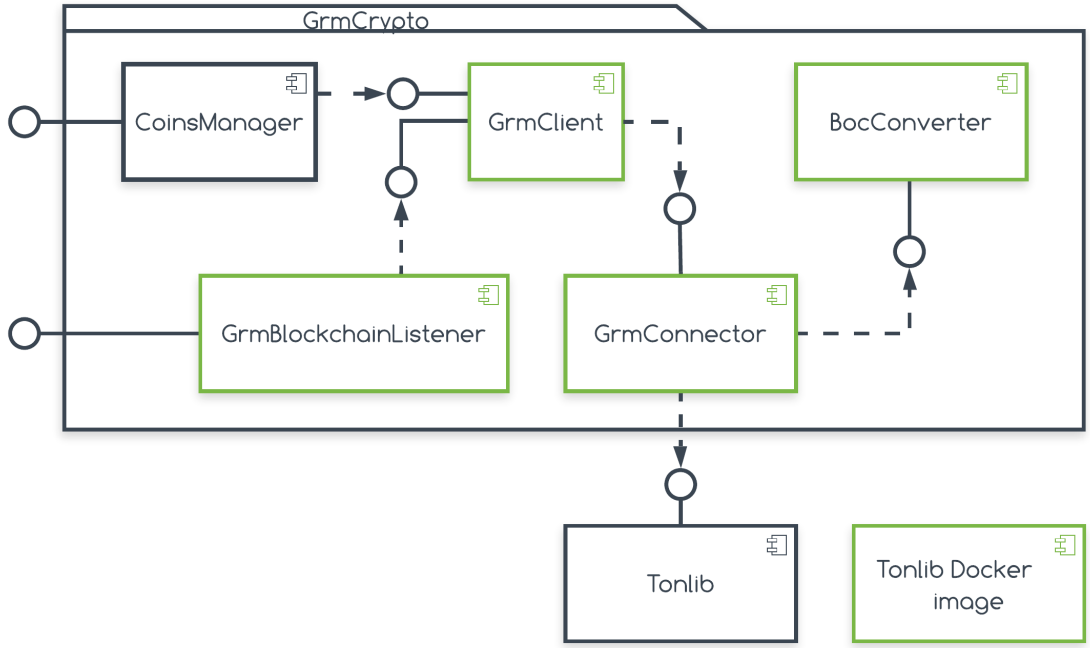

Рис. 3: Модуль поддержки криптовалюты Gram

5.1. TON API

Взаимодействие с платформой TON из сторонних приложений осуществляется с помощью библиотеки tonlib[21], которая написана на C++14. В официальном репозитории платформы существует пример[22] использования возможностей библиотеки, другой вид документации отсутствует. Помимо библиотеки tonlib существует её JNI [12] обвязка для вызова функций библиотеки из Java кода. Java Native Interface (JNI) — это механизм запуска кода под управлением виртуальной машины Java, который написан на языках C, C++ или Ассемблере и скомпилирован в виде динамической библиотеки.

[21]https://github.com/ton-blockchain/ton/tree/master/tonlib

[22]https://github.com/ton-blockchain/ton/tree/master/example/android/test/ton/src/androidTest/java/drinkless/org/ton

Взаимодействие с узлами сети blockchain происходит по протоколам ADNL и RLDP [3]. Эти протоколы специально разработаны для платформы TON и не имеют документации. В связи с этим, для взаимодействия с узлом сети TON Blockchain используется библиотека tonlib.

Процесс сборки библиотеки tonlib и генерации её обвязки не документирован, содержит несколько этапов и требует специальные пакеты и версии программного обеспечения. Чтобы упростить и автоматизировать процесс сборки был создан Doker-образ[23], который содержит все необходимое ПО, по скрипту собирает tonlib и генерирует JNI обвязку.

5.2. TON Blockchain

5.2.1. Account model

TON Blockchain использует модель аккаунта с балансом. Каждый аккаунт представляет собой смарт-контракт. Смарт-контракт — это сущность в blockchain, которая имеет адрес, состояние и некоторый код. Взаимодействие со аккаунтом происходит с помощью сообщений, которые подразделяются на внешние, отправленные в blockchain из реального мира, и внутренние, отправленные смарт-контрактами. Сообщения содержат адрес отправителя, адрес получателя, количество валюты, прикрепленной к этому сообщению, а также произвольные данные, которые интерпретируются аккаунтом-получателем. Каждая транзакция аккаунта содержит одно обработанное входящее сообщение, изменения состояния аккаунта и сгенерированные исходящие сообщения, которых может быть несколько. TON Blockchain гарантирует, что сообщение будет доставлено только один раз. За каждое отправленное сообщение в TON Blockchain взимается комиссия.

Модель аккаунта с балансом в TON Blockchain представлена на рис. 4.

Комиссия взимается не только за отправку сообщений, но и за хранение данных в смарт-контрактах. Такая плата взимается при получении

[23]https://github.com/dsx-tech/Blockchain-Payment-System/tree/master/src/main/resources/Docker%20images/ton-nativelib-image

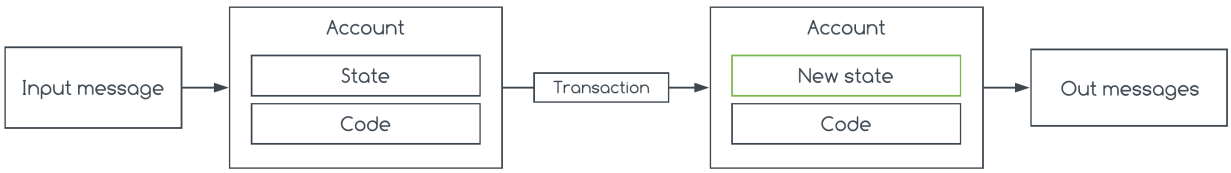

Рис. 4: Модель аккаунта с балансом в TON Blockchain

смарт-контрактом сообщения и зависит от количества новых блоков в blockchain, начиная с блока с последней транзакцией аккаунта. Если смарт-контракт не имеет достаточно средств для оплаты комиссии, он замораживается. В таком состоянии смарт-контракт можно восстановить в рабочее состояние. Если аккаунт не восстановить, то через некоторое время он полностью удаляется из blockchain. Комиссия за хранение данных защищает сеть от нежелательного быстрого роста объема хранящихся данных в ней, делая создание большого количества аккаунтов невыгодным. Именно поэтому, для ввода и вывода криптовалюты Gram в платежной системе BPS используется один аккаунт в TON Blockchain. Для отслеживания платежей используются произвольные идентификаторы, которые содержатся в данных сообщения.

Для защиты от повторного списания средств смарт-контракты имеют специальный механизм. Состояние каждого контракта содержит специальное число — sequence number. В каждом сообщении указывается sequence number аккаунта получателя. Если при обработке сообщения эти два числа не совпадают, то обработка сообщения завершается ошибкой. После успешной обработки сообщения sequence number смарт-контракта автоматически увеличивается на единицу. Таким образом, если одно и то же сообщение попытаться обработать два раза, то во второй раз sequence number сообщения и смарт-контракта не совпадут, обработка завершится с ошибкой и повторного списания средств не произойдет.

5.2.2. Infinite Sharding Paradigm

TON Blockchain имеет встроенный механизм масштабирования Infinite Sharding Paradigm [3] [5]. Всё множество возможных адресов аккаунтов

разбивается на непересекающиеся подмножества. Транзакции каждого подмножества аккаунтов обрабатываются в отдельном blockchain, который называется шардчейном (shardchain). Разбиение аккаунтов на подмножества происходит по бинарному префиксу их адресов. Таким образом, транзакции аккаунтов, адреса которых имеют одинаковый префикс, обрабатываются в одном шардчейне.

Как только в шардчейн начинает поступать черезмерно много транзакций для обработки, он расщепляется на два дочерних шардчейна — их префиксы становятся на один бит длиннее родительского. Если шардчейн в течение некоторого времени обрабатывает мало транзакций, то он автоматически сливается с шардчейном, префикс которого такой же длины и отличается на один последний бит. Таким образом, шардчейны помогают балансировать нагрузку и повышают пропускную способность всей системы, когда это необходимо. Механизм расщепления и слияния шардчейнов представлен на рис. 5

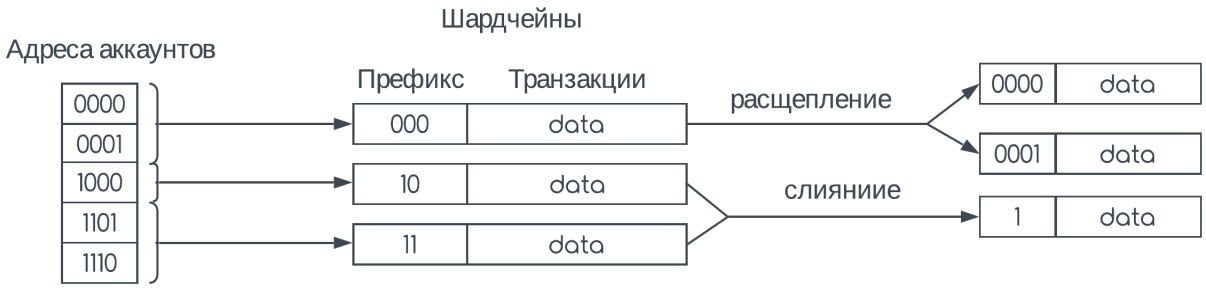

Рис. 5: Механизм расщепления и слияния шардчейнов

Для поддержания согласованности всей системы существует blockchain — мастерчейн (masterchain), в котором фиксируются состояния шардчейнов. В его блоках хранится количество и префиксы шардчейнов, хеши последних добавленных блоков в шардчейны, а также другие данные, необходимые для обеспечения согласованности.

Помимо механизма масштабирования, шардчейны имеют еще одно важное свойство. Валидаторы — узлы сети, которые занимаются формированием новых блоков и получают за них вознаграждения. Валидаторы прикрепляются к определенному шардчейну и сменяются каждый час. Для того, чтобы работать с транзакциями аккаунтов определенно-

го шардчейна, валидаторам необходимо хранить данные только этого шардчейна, а не всей системы. Таким образом, требования к оборудованию валидаторов значительно снижаются.

5.3. Отслеживание транзакций

TON Blockchain использует смесь протоколов консенсуса Proof-of-Stack (PoS) и Byzantine Fault Tolerant (BFT) — Catchain Consensus [4].

Процесс выбора валидаторов происходит по протоколу PoS. Каждый участник сети, желающий стать валидатором, резервирует некоторую сумму криптовалюты и получает возможность стать валидатором. Вероятность того, что участник станет валидатором зависит от доли, которую составляют средства, замороженные участником, к общему количеству средств, замороженных всеми участниками. Зарезервированные средства участника называются его стейком. Новые валидаторы выбираются каждый месяц. Стейк валидаторов остается зарезервированным на протяжении всего срока валидации и на один месяц после его окончания. Весь процесс выбора валидаторов осуществляет специальный смарт-контракт.

Решение о включении транзакции в блок принимается валидаторами по протоколу BFT [3] [7]. Транзакция включается в следующий блок, если более 2/3 валидаторов подтверждают ее включение. Благодаря такому подходу, отпадает необходимость в ожидании определенного количества подтверждений после включения транзакции в новый блок, как это происходит в Bitcoin.

Catchain Consensus теоретически допускает ситуацию, при которой невалидный блок будет принят сетью. Для решения этой проблемы предусмотрена процедура исправления некорректного блока и всей последующей цепочки блоков. Поверх невалидных блоков принимаются корректирующие блоки, которые восстанавливают корректность всего blockchain. Таким образом над некорректным блоком основного blockchain образуется "вертикальный" blockchain корректирующих блоков. Процедура восстановления блока допускается только в течение месяца с мо-

мента принятия блока сетью. Валидаторы, принявшие некорректный блок, отстраняются от валидации новых блоков и теряют часть стейка. Таким образом, валидаторам невыгодно принимать некорректные блоки.

Принцип исправления некорректных блоков представлен на рис. 6

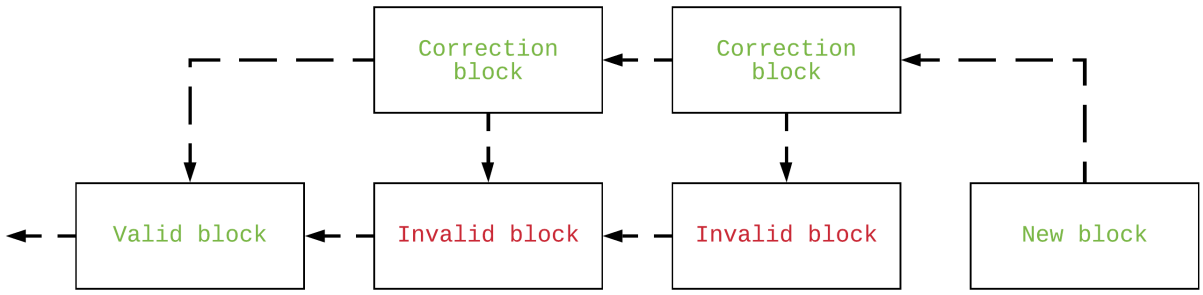

Рис. 6: Исправление некорректных блоков

В рамках платежной системы BPS корректно обрабатывать ситуации восстановления невалидных блоков не представляется возможным. Помимо этого, учитывая наказание недобросовестных валидаторов, вероятность исправления блоков очень мала, поэтому при мониторинге транзакций считаем, что если транзакция включена в блок, который принят сетью, то она уже не будет отменена.

5.4. Отслеживание платежей

5.4.1. Процесс ввода средств

Процесс ввода средств в платежную систему построен следующим образом. Сначала создается объект Invoice, который содержит все данные об ожидаемом поступлении средств на аккаунт платежной системы. Далее генерируется специальный идентификатор, который должен указать в данных отправитель сообщения перевода криптовалюты. После этого начинается мониторинг сети на наличие транзакций аккаунта платежной системы, которые обрабатывают входящее сообщение с заданным идентификатором. Все найденные транзакции проверяются и вычисляется общая сумма поступивших средств. Как только Invoice

полностью оплачен, выставляется соответствующий статус и процесс завершается.

Процесс ввода криптовалюты Gram в платежную систему BPS представлен на рис. 7

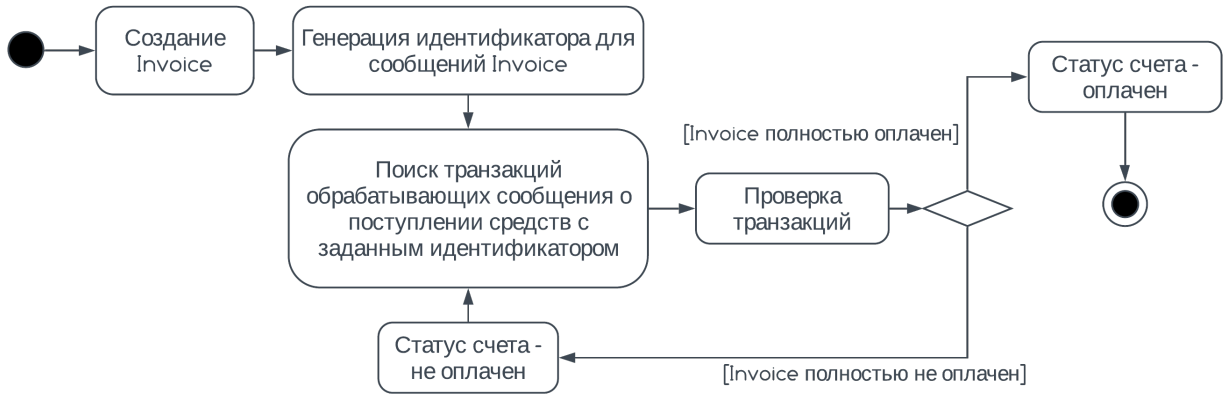

Рис. 7: Процесс ввода Gram в платежную систему BPS

5.4.2. Процесс вывода средств

Процесс вывода средств с платежной системы устроен следующим образом. Вначале создается объект Payment, которые содержит всю информацию о выводе средств. Далее отправляется сообщение о выводе средств на аккаунт платежной системы в blockchain. Идентификатор отправленного сообщения записывается в Payment. После отправки сообщения начинается мониторинг сети на наличие транзакции, которая обрабатывает отправленное сообщение. Если транзакция найдена, то проверяется, что обработка сообщения прошла успешно и сообщение с необходимым количеством криптовалюты отправлено на целевой адрес. После этого обновляется статус платежа.

Процесс вывода криптовалюты Gram из платежной системы BPS представлен на рис. 8

Важно отметить, что при выводе средств платежная система не проверяет, что сообщение с криптовалютой корректно обработано на целевом аккаунте и оставляет эту задачу получателю перевода. Таким образом, платежная система хранит только данные шардчейна, в котором обрабатывается аккаунт платежной системы. Если проверять кор-

Рис. 8: Процесс вывода Gram из платежной системы BPS

ректность обработки отправленного сообщения на целевом аккаунте, то необходимо хранить данные всех шардчейнов сети, что критично повышает требования к оборудованию для работы BPS и делает использование платежной системы практически недоступным.

5.4.3. Идентификация платежей

Для отслеживания платежей используются шестнадцатеричные строковые идентификаторы в кодировке UTF-8, которые указываются в данных сообщения. Данные сообщения хранятся в специальном формате — bag of cells [3]. Bag of cells (BoC) — формат хранения данных представляющий собой дерево, каждый узел которого является cell. Cell — структура данных, которая содержит от 0 до 128 байт сырых данных и от 0 до 4 ссылок на другие cell. Спецификация BoC описана в формате tlb [24]. Для обеспечения корректного отслеживания платежей необходимо правильно извлекать строковые идентификаторы из формата BoC. TON API обрабатывает данные в формате BoC следующим образом:

- Если перед данными установлены ведущие 4 байта со значением

[24]https://github.com/ton-blockchain/ton/blob/master/crypto/tl/boc.tlb

0, то в этом случае API обходит BoC в глубину и представляет данные сообщения как строку UTF-8.

- Если перед данными установлены ведущие 4 байта со значением 255, то в этом случае BoC зашифрован. API расшифровывает BoC и обходит его в глубину, представляя данные как строку UTF-8.

- В остальных случаях API возвращает данные сообщения в виде сериализованного BoC.

Для корректной обработки последнего случая написан конвертер данных из сериализованного формата BoC в строковый формат в кодировке UTF-8.

Заключение

В рамках курсовой работы были выполнены следующие задачи:

- Сделан обзор предметной области
- Проведен анализ существующих решений
- Доработан прототип платежной системы
- Поддержана криптовалюта Gram
- Проведена апробация поддержки криптовалюты Gram на тестовой сети

Благодарности

Я хочу поблагодарить компанию DSX Technologies за возможность получить актуальные знания и применить их на практике, а также за финансовую поддержку во время работы над курсовой работой. Особенно хочу выразить благодарность техническому консультанту, Филиппу Долголеву, за эффективное руководство и профессиональные советы, а также поблагодарить Сергея Скаредова за консультации по кодовой базе и код-ревью.

Список литературы

[1] Blockchain use-cases. — 2019. — URL: https://consensys.net/enterprise-ethereum/use-cases/ (дата обращения: 27.11.2019).

[2] CoinMarketCup. — 2019. — URL: https://coinmarketcap.com (дата обращения: 27.11.2019).

[3] Durov Nikolai. Telegram Open Network. — 2019. — URL: https://test.ton.org/ton.pdf (дата обращения: 27.11.2019).

[4] Durov Nikolai. Catchain Consensus: An Outline. — 2020. — URL: https://test.ton.org/catchain.pdf (дата обращения: 20.04.2020).

[5] Durov Nikolai. Telegram Open Network Blockchain. — 2020. — URL: https://test.ton.org/tblkch.pdf (дата обращения: 06.03.2020).

[6] Information technology – Programming languages – C++ : Standard : ISO/IEC 14882:2014 / International Organization for Standardization ; Executor: ISO Central Secretary. — Geneva, CH : 2014. — dec.

[7] Lamport Leslie, Shostak Robert, Pease Marshall. The Byzantine Generals Problem // ACM Transactions on Programming Languages and Systems. — 1982. — July. — P. 382–401. — URL: https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

[8] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. — 2008. — URL: https://bitcoin.org/bitcoin.pdf (дата обращения: 27.11.2019).

[9] Schollmeier Rüdiger. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. — 2001. — 09. — P. 101 – 102.

[10] Wahab Abdul, Mehmood Waqas. Survey of Consensus Protocols // CoRR. — 2018. — Vol. abs/1810.03357. — 1810.03357.

[11] Wikipedia contributors. 3-D Secure — Wikipedia, The Free Encyclopedia. — 2019. — URL: https://en.wikipedia.org/w/index.php?title=3-D_Secure&oldid=924575552 (дата обращения: 27.11.2019).

[12] Wikipedia contributors. Java Native Interface — Wikipedia, The Free Encyclopedia. — 2019. — [Online; accessed 19-April-2020]. URL: https://en.wikipedia.org/w/index.php?title=Java_Native_Interface&oldid=933299999.

[13] Унификация взаимодействия с различными реализациями blockchain в качестве платежных систем. — 2019. — URL: http://se.math.spbu.ru/SE/YearlyProjects/spring-2019/371/Skaredov-report.pdf (дата обращения: 27.11.2019).