Санкт-Петербургский государственный университет

Кафедра системного программирования
Программная инженерия

Скаредов Сергей Антонович

Унификация взаимодействия с различными реализациями blockchain в качестве платежных систем

Курсовая работа

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

Санкт-Петербург
2019

Оглавление

Введение

Криптовалюта — цифровой актив, в основу которого заложены принципы криптографии и отсутствия централизованного контролирующего аппарата. С момента публикации статьи [9] о реализации первой криптовалюты Bitcoin в 2008 году на момент мая 2019 года мировой рынок насчитывает свыше 2000 различных криптовалют [7]. Многие из них, как и Bitcoin, базируются на технологии blockchain — распределённой децентрализованной базе данных — описанной в той же статье. Она позволяет надёжно хранить историю всех операций в сети и обеспечивает строго консистентное изменение данных.

Перечисленные особенности криптовалют с каждым годом делают их всё более популярными. Появляются новые криптовалютные биржи, увеличивается количество организаций, связывающих свою деятельность с миром криптовалют.

В связи с активным развитием данного направления и обилием различных реализаций blockchain в качестве платёжных систем, появляется необходимость в инструменте, который позволит взаимодействовать с разными криптовалютами по одним и тем же принципам. Это будет актуально для банков, торговых компаний, индивидуальных предпринимателей, рассматривающих возможность использования криптовалюты для предоставления своих услуг, продажи товаров и других финансовых операций.

Прямую аналогию можно провести с платёжными системами, которые скрывают под уровнем абстракции детали реализации работы с различными банками и предоставляют пользователю удобный интерфейс выставления счетов, приёма и отправки платежей.

1. Постановка задачи

Целью данной работы является разработка системы, позволяющей взаимодействовать с различными криптовалютами на основе blockchain, производить стандартные операции участникам финансовых отношений. Для достижения цели были поставлены следующие задачи:

- Провести анализ существующих решений
- Спроектировать архитектуру системы
  - Разработать модуль, отвечающий за выполнение стандартных финансовых операций
  - Разработать унифицированный интерфейс для работы с криптовалютой вне зависимости от её конкретной реализации
- Реализовать прототип системы
- Реализовать поддержку популярных криптовалют
- Провести апробацию системы

2. Обзор предметной области

2.1. Blockchain

Технология blockchain представляет собой распределённую децентрализованную базу данных. Каждый участник сети хранит копию истории операций — цепочку последовательных блоков. Блок состоит из набора транзакций, идентификатора предыдущего блока и некоторых дополнительных данных. Идентификатор блока — хеш от содержащихся в блоке данных. Это значит, что изменение даже одной транзакции в уже принятом сетью блоке изменит его идентификатор и, следовательно, сделает невалидной всю оставшуюся цепочку. Такая связь блоков защищает сеть от попыток изменения истории.

Область применения blockchain включает логистику, распространение контента, системы голосования, интернет вещей и другие сферы [17], но в рамках данной работы рассмотрены только реализации blockchain, относящиеся к криптовалютам.

2.2. Консенсус

Децентрализация в принятии решений, то есть отсутствие третьей доверенной стороны, контролирующей процесс проведения операций в сети, достигается применением протокола консенсуса для валидации транзакций. Существуют различные протоколы консенсуса, такие как Proof-of-Work, Proof-of-Stake, Practical Byzantine Fault Tolerance и другие [16]. Особенности используемого протокола влияют на такие характеристики сети blockchain, надёжность и устойчивость к различного рода атакам, а так же на количество транзакций в секунду, время генерации нового блока, способ подтверждения того, что транзакция была принята сетью.

3. Обзор существующих аналогов

В настоящее время существуют решения, которые предоставляют криптовалютные платёжные шлюзы. Такие сервисы выгодны для предпринимателей, которые намерены поддержать тренды в сфере финансов и повысить лояльность клиентов к себе, но не имеют ресурсов на освоение новой специфичной для них технологии. В таблице 1 представлены некоторые популярные сервисы [1] для криптовалютных платежей.

Сервис	Валюты	Кастодиальный	Хостинг	Open source
bitpay[1]	2	+	+	-
coingate[2]	>50	+	+	-
CoinPayments[3]	>70	+	+	-
paycoiner[4]	18	+	+	-
Coinbase Commerce[5]	4	-	+	-
BTCPay Server[6]	11	-	-	+

Таблица 1: Популярные криптовалютные платёжные системы

Параметры для сравнения были выбраны с целью проанализировать сервисы на объём поддерживаемых криптовалют, способ хранения средств и доступность сервисов для рядового пользователя.

Чаще подобное программное обеспечение является проприетарным, что не позволяет рассмотреть детали реализации архитектуры такого продукта. Кастодиальные сервисы хранят средства пользователей в своих собственных кошельках, предоставляя возможность вывода активов с определённой комиссией. Некастодиальные же переводят валюту напрямую в кошельки пользователей. Снятие необходимости хостинга программного обеспечения с пользователей так же влечёт за собой взятие некоторой платы. Продукт BTCPay Server является

[1]https://bitpay.com/
[2]https://coingate.com/
[3]https://www.coinpayments.net/
[4]https://paycoiner.com/
[5]https://commerce.coinbase.com/
[6]https://btcpayserver.org/

некастодиальным self-hosted проектом с открытым исходным кодом, но поддерживает работу только с Bitcoin и его fork-производными.

4. Разработка архитектуры

Возможность распоряжаться своими средствами не должна зависеть от специфики работы с конкретной криптовалютой. Поэтому было принято решение разделить структуру сервиса на два функциональных модуля. Модуль Core отвечает за финансовые операции в контексте платежей и счетов. Модуль Crypto осуществляет взаимодействие с различными реализациями blockchain.

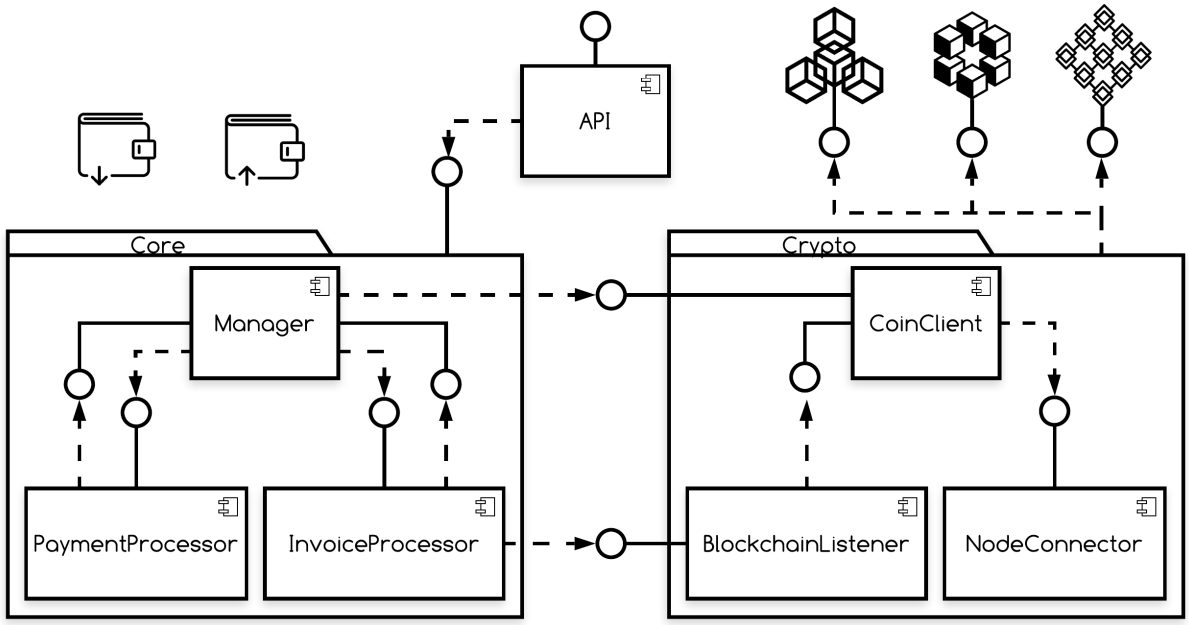

Рис. 1: Архитектура системы

Такая архитектура (рис. 1) позволит легко масштабировать систему, добавляя поддержку новых криптовалют во второй блок, в то время как первый будет взаимодействовать с реализацией для каждой валюты по одному принципу.

Для разработки проекта был выбран язык программирования Kotlin. Он совместим с языком Java, но позволяет писать более компактный код и обладает широким набором вспомогательных библиотек как собственных, так и из мира Java-разработки. В том числе многие blockchain-платформы предоставляют open source библиотеки на этих языках для взаимодействия с ними.

4.1. Модуль Core

Модуль Core включает три сервиса и оперирует сущностями платёж (Payment), счёт (Invoice) и транзакция (Tx).

4.1.1. PaymentProcessor

PaymentProcessor по запросу создаёт и сохраняет объект Payment с параметрами идентификатора криптовалюты, её количества и адреса получателя. Затем Payment передаётся в соответствующий blockchain модулем Crypto. После того, как транзакция отправлена, её идентификатор записывается в объект Payment. С его помощью PaymentProcessor может запрашивать информацию о транзакции в виде объекта Tx и обновлять статус платежа.

4.1.2. InvoiceProcessor

InvoiceProcessor отвечает за создание объектов Invoice и отслеживание их статуса.

Invoice создаётся с параметрами уникального идентификатора счёта, идентификатора криптовалюты, её количества и адреса, на котором ожидается получение средств. InvoiceProcessor подписывается на получение новых транзакций в виде объектов Tx, которые публикуются модулем Crypto. Сравнивая соответствующие параметры новых транзакций и выставленных счетов, InvoiceProcessor находит Invoice, оплата которого была произведена в транзакции, и обновляет количество полученных средств и статус счёта.

В некоторых реализациях blockchain существует возможность ветвления цепочки блоков. Например, в консенсусе Proof-of-Work майнеры — узлы сети, генерирующие блоки — могут почти в один момент времени произвести два разных блока, которые будут разосланы по сети [4]. Это значит, что даже валидная транзакция может быть отменена, пока не наберёт рекомендуемое количество подтверждений — новых блоков цепи, сгенерированных после блока, который содержит транзакцию. Поэтому идентификатор транзакции,

в которой произошло начисление валюты по выставленному счёту, сохраняется в объект Invoice. Это позволит в дальнейшем проверять статус входящей транзакции и поддерживать актуальным статус счёта.

4.1.3. Manager

Manager координирует работу всей системы. Он инициирует создание платежей и счетов, взаимодействует с сетями blockchain через интерфейс, предоставляемый модулем Crypto, а также перенаправляет запросы транзакций от PaymentProcessor и InvoiceProcessor в модуль Crypto.

4.2. Модуль Crypto

С целью сделать поддержку новых криптовалют проще, в процессе разработки схожие аспекты взаимодействия с разными сетями blockchain были вынесены и обобщены. Таким образом, модуль Crypto состоит из наборов реализации трёх сервисов для каждой криптовалюты.

4.2.1. NodeConnector

Узлы сети Bitcoin предоставляют JSON-RPC API [11], что было перенято его последователями и стало стандартом де-факто в мире blockchain. Каждая сеть обладает уникальным набором методов, но процессы отправки запросов и получения ответов следуют одним принципам.

В связи с этим в сервис NodeConnector была вынесена функциональность по установлению соединения с узлом сети и отправке запросов. Также представлена базовая функциональность по формированию запросов и разбору ответов, но непосредственные реализации сервиса, специфичные для конкретной валюты, могут переопределять их. Такая абстракция позволит взаимодействовать и с иными видами API, но рассмотренные в этой работе криптовалюты поддерживают именно JSON-RPC.

При поддержке криптовалюты необходимо реализовать данный сервис, поддерживающий её API.

4.2.2. CoinClient

Сервис CoinClient описывает интерфейс, с помощью которого модуль Core взаимодействует с blockchain. Например, запрос баланса, запрос транзакции или отправка платежа. Используя соответствующий NodeConnector для общения с узлом, реализация CoinClient является посредником между контекстом своей криптовалюты и упрощённым финансовым контекстом системы. Так, например, каждый CoinClient реализует функциональность по конвертации специфичных для blockchain типов транзакций в обобщённое представление транзакций — объект типа Tx.

4.2.3. BlockchainListener

Для отслеживания начислений по выставленным счетам требуется непрерывное наблюдение за транзакциями в сети. Сервис BlockchainListener объявляет методы мониторинга и публикации новых транзакций, которые нужно проверить.

Количество транзакций в секунду сильно разнится от реализации к реализации [5], а постоянное развитие технологии blockchain непрерывно увеличивает это число, что является критичным моментом при мониторинге. Большая часть операций в сети никак не связана с одним аккаунтом пользователя. Поэтому каждая имплементация сервиса BlockchainListener должна учитывать особенности blockchain, с которым она работает, и разумно использовать возможности, предоставляемые API узлов.

5. Поддержка криптовалют

Поддержка определённой цифровой монеты требует изучить некоторые детали её реализации:

- API для взаимодействия с blockchain
- Принципы идентификации пользователей сети
- Способы формирования транзакций
- Способы валидации совершённых платежей

Для работы над проектом были выбраны три популярные криптовалюты: Bitcoin (BTC), Ripple (XRP) и Tron (TRX). Они входят в топ-11 по объёму рыночной капитализации [7], а их реализации сильно отличаются друг от друга, что позволит должным образом обобщить схожую функциональность для упрощения дальнейшей разработки и поддержки новых криптовалют.

Валюта	Капитализация ($)	Протокол	Время блока	TPS[7]
Bitcoin	128.4 млрд	PoW	10-15 мин	3-6
Ripple	13.3 млрд	XRP LCP	3 сек	6 / 1500
Tron	1.6 млрд	DPoS	3 сек	20 / 2000

Таблица 2: Поддерживаемые криптовалюты

5.1. Bitcoin

5.1.1. Unspent Transaction Output (UTXO)

Bitcoin использует модель учёта средств Unspent Transaction Output (UTXO). Это значит, что баланс участника сети есть сумма непотраченных средств из входящих для него транзакций — непотраченных выходов.

Для каждой транзакции определяются выходы — адреса получателей и количество валюты, а также входы — непотраченные

[7]TPS — Transaction per Second

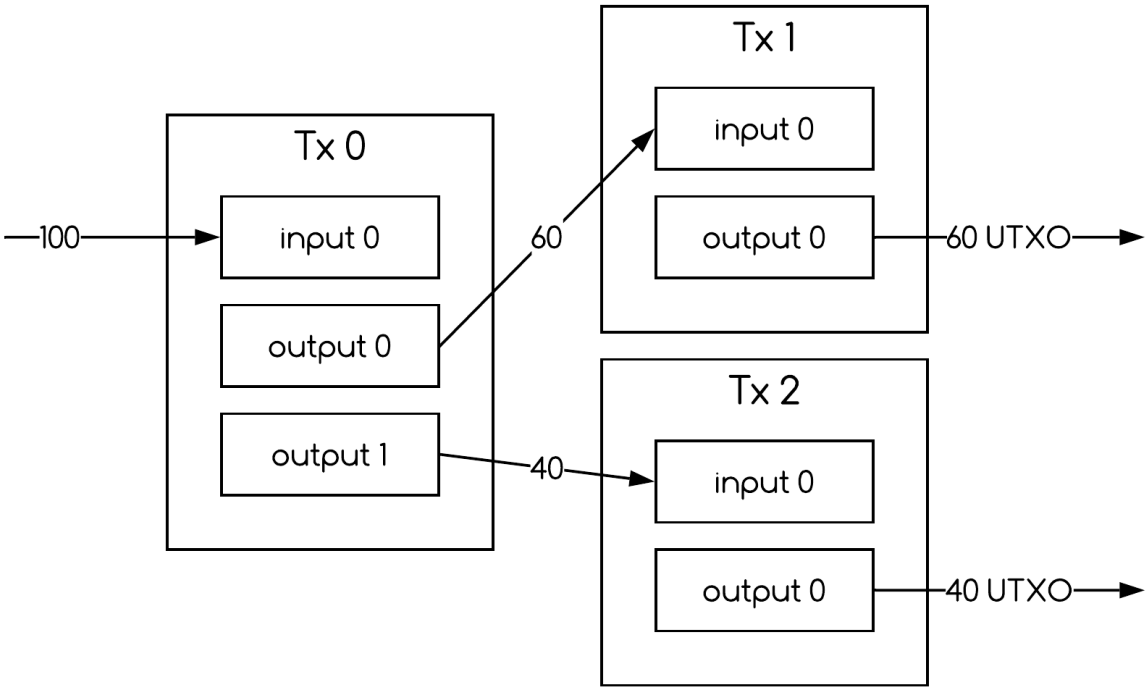

Рис. 2: Unspent Transaction Output (UTXO)

выходы, средства которых будут переведены. Для того, чтобы воспользоваться каким-то UTXO, необходимо обладать приватным ключём, который соответствует адресу получателя, указанному в UTXO. Таким образом кошелёк пользователя представляет из себя набор приватных ключей, из которых возможно получать публичные ключи-адреса. Разработчики Bitcoin настаивают [2] на использовании новых адресов для получения средств с целью повышения анонимности участников сети. Такая особенность сети хорошо подходит для выставления счетов на оплату, позволяя легко дифференцировать входящие платежи.

5.1.2. Мониторинг

Bitcoin использует протокол Proof-of-Work [10]. В силу того, что сразу несколько майнеров пытаются создать и записать новый блок, возможны ситуации ветвления. Поэтому документация Bitcoin предусматривает рекомендуемое число подтверждений, но

пользователи сети в праве для себя решать, какое количество новых блоков будет достаточным. В связи с этим хранение идентификатора входящей или исходящей транзакции, а также проверка её статуса продолжаются до набора необходимого пользователю количества подтверждений.

В случае, если входящая транзакция оказалась в ответвлении цепочки, которое по итогу было отвергнуто, баланс отправителя не изменяется, а счёт продолжает ожидать оплаты. Если исходящая транзакция оказывается в подобной ситуации, платёж следует повторить, либо отменить, следуя указаниям пользователя.

Сеть Bitcoin обладает низкой пропускной способностью. В настоящий момент сеть производит один блок в среднем каждые десять минут [3]. Значит, для подтверждения транзакции необходим один час. TPS — от трёх до шести, что очень мало в сравнении с высокопроизводительными сетями, такими как EOS или Tron [5]. К тому же API узлов Bitcoin предоставляет методы для получения только тех транзакций, которые относятся к кошельку пользователя, что упрощает процесс мониторинга.

API Bitcoin-узлов даёт возможность формировать транзакции, не указывая какие конкретно UTXO использовать для перевода средств. При таком запросе узел сети сам подбирает непотраченные выходы пользователя, что позволяет абстрагироваться от деталей и упрощает поддержку данной валюты в рамках проекта.

5.2. Ripple

XRP Ledger — распределённая платёжная система, разработанная для доступных и быстрых платежей по всему миру. Система поддерживает переводы различных цифровых активов и фиатных валют, а также их обмен, но в рамках данной работы рассмотрена только основная криптовалюта XRP.

5.2.1. Account model

Наличие в блоках Ripple информации об изменении состояний аккаунтов позволяет отказаться от концепции UTXO. Так, в отличие от Bitcoin, кошелёк пользователя Ripple представлен аккаунтом с одним адресом и балансом [12]. Каждый аккаунт должен содержать "резервное" количество XRP. Это защищает сеть от нежелательно быстрого роста хранящихся в ней данных, так как делает затратным создание и использование новых адресов. Для отслеживания платежей в таком случае используются дополнительные данные, которые указываются в транзакциях, например, "DestinationTag" [14].

5.2.2. Мониторинг

Ripple использует XRP Ledger Consensus Protocol [6, 8], при котором валидаторы — узлы сети, которые занимаются формированием новых блоков — коллегиально принимают решение, какие транзакции будут или не будут включены в новый блок. Благодаря такому подходу отпадает необходимость в ожидании определённого количества подтверждений. Если транзакция включена в блок, принятый сетью, она уже не будет отменена.

Среднее время генерации нового блока три секунды. Около шести транзакций появляется в сети за секунду с возможным ростом до 1500 TPS [13]. API узлов Ripple, как и Bitcoin, позволяет запрашивать транзакции, относящиеся только к аккаунту пользователя.

5.3. Tron

Tron — децентрализованная blockchain-платформа с возможностью исполнения смарт-контрактов для хранения, распространения и монетизации контента. Подобно платформе Ethereum, позволяет создавать и использовать "токены" [15], но в рамках данного проекта рассмотрена работа только с основной валютой платформы — TRX.

5.3.1. Account model

Tron, как и Ripple, использует модель аккаунта с балансом. Консенсус Delegated Proof-of-Stake, в соответствии с которым работает сеть Tron, подразумевает, что каждый пользователь в праве заморозить некоторую часть своих средств с целью участвовать в голосовании за узлы, которые будут производить блоки до следующего голосования [15]. Эта функциональность сети, в отличие от возможности перевода валюты, не поддержана в рамках проекта, так как не относится к общему финансовому контексту проекта. Но в силу того, что баланс аккаунта может изменяться в зависимости от действий пользователя вне разрабатываемой системы, необходимо запрашивать информацию об актуальном количестве доступных на счету средств для каждого платежа из системы.

5.3.2. Мониторинг

Количество подтверждений транзакции, рекомендуемое документацией Tron — 19 новых блоков. Новый блок генерируется каждые три секунды, а заявленная пропускная способность сети — 2000 TPS. Это больше, чем прежде рассмотренные варианты, к тому же API узлов Tron не содержит методов, позволяющих запрашивать транзакции, относящиеся исключительно к аккаунту пользователя. В силу ограниченности API, процесс мониторинга данного blockchain заключается в последовательном переборе всех транзакций из каждого нового блока, отсеивая те, что не являются входящими или исходящими.

Заключение

В рамках данной работы были достигнуты следующие результаты:

- Изучена предметная область
- Проведён анализ существующих решений
- Реализован прототип системы
- Поддержана криптовалюта Bitcoin
- Поддержана криптовалюта Ripple
- Поддержана криптовалюта Tron
- Проведена апробация системы на тестовых сетях blockchain

Список литературы

[1] 10 Best Bitcoin Payment Gateways for 2019. — 2019. — URL: https://www.devteam.space/blog/10-best-bitcoin-payment-gateways-for-2019/ (дата обращения: 06.05.2019).

[2] Avoiding Key Reuse. — 2009. — URL: https://bitcoin.org/en/transactions-guide#avoiding-key-reuse (дата обращения: 10.05.2019).

[3] Bitcoin Explorer. — 2019. — URL: https://live.blockcypher.com/btc/ (дата обращения: 11.05.2019).

[4] Block Height And Forking. — 2009. — URL: https://bitcoin.org/en/blockchain-guide#block-height-and-forking (дата обращения: 06.05.2019).

[5] Blockchain Activity Matrix. — 2019. — URL: https://www.blocktivity.info (дата обращения: 11.05.2019).

[6] Chase Brad, MacBrough Ethan. Analysis of the XRP Ledger Consensus Protocol // CoRR. — 2018. — Vol. abs/1802.07242. — 1802.07242.

[7] Coinmarketcap. — 2018. — URL: https://coinmarketcap.com (дата обращения: 06.05.2019).

[8] David Schwartz Noah Youngs Arthur Britto. The Ripple Protocol Consensus Algorithm. — 2014. — URL: https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[9] Nakamoto Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System. — 2008. — URL: https://bitcoin.org/bitcoin.pdf.

[10] Proof Of Work. — 2009. — URL: https://bitcoin.org/en/blockchain-guide#proof-of-work (дата обращения: 10.05.2019).

[11] Remote Procedure Calls (RPCs). — 2009. — URL: https://bitcoin.org/en/developer-reference#remote-procedure-calls-rpcs (дата обращения: 06.05.2019).

[12] Ripple Accounts. — 2019. — URL: https://developers.ripple.com/accounts.html (дата обращения: 10.05.2019).

[13] Ripple Market Performance. — 2019. — URL: https://ripple.com/xrp/market-performance (дата обращения: 10.05.2019).

[14] Ripple Payment Transaction. — 2019. — URL: https://developers.ripple.com/payment.html (дата обращения: 10.05.2019).

[15] Tron. Advanced Decentralized Blockchain Platform. — 2018. — URL: https://tron.network/static/doc/white_paper_v_2_0.pdf.

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

[17] Zīle Kaspars, Strazdiņa Renāte. Blockchain Use Cases and Their Feasibility // Applied Computer Systems. — 2018. — 05. — Vol. 23. — P. 12–20.