Блокчейн UNB

1. Общая информация о блокчейне

1.1. Блокчейн - система состояний

Как известно понятие блокчейн появилось в 2008 году с созданием Bitcoin. Протокол Bitcoin представляет собой систему изменения его состояния, которое описывает то, как распределены все существующие биткойны между различными адресами. Bitcoin имеет ограниченное применение, он предназначен для использования исключительно как децентрализованная цифровая валюта. Другие децентрализованные приложения, такие как организация децентрализованного хранения информации или построение умных контрактов, крайне неэффективны на Bitcoin. Для решения этих проблем был придуман Ethereum, который открыл следующий класс блокчейнов - со встроенными смарт контрактами.

Блокчейн UNB является форком Ethereum и унаследовал все его преимущества, улучшив показатели быстродействия и оптимизировав его для работы в управляемых (закрытых) системах.

Блокчейн UNB является системой состояний (или «машина состояний»), т.е. системой, которая обрабатывает вводимую информацию и преобразуется в новое состояние. Состояние блокчейна UNB определяется как суперпозиция состояний его аккаунтов, функция изменения состояния — передача информации между аккаунтами.

1.2. Аккаунты в блокчейне

Аккаунт в блокчейне UNB - это учетная запись, соответствующая паре криптографических ключей: открытому и закрытому. Открытый ключ выступает в качестве исходных данных для генерации адреса аккаунта (который для краткости часто также

называют аккаунтом), закрытый ключ используется для формирования электронной подписи сообщений с данного аккаунта.

Каждому аккаунту UNB соответствует его 20-байтовый адрес. Аккаунт UNB содержит четыре поля:

- nonce — счетчик транзакций, необходимый для предотвращения дублирования транзакций;
- код смарт контракта, связанного с аккаунтом (при наличии смарт контракта);
- хранилище аккаунта);
- текущий баланс аккаунта (служебное поле для совместимости с Ethereum - например в ETH).

В блокчейне UNB возможны два типа аккаунтов: контролируемый владельцем аккаунт (externally owned account) и аккаунт, контролируемый кодом связанного с ним смарт контракта (контракт-аккаунт). Первый тип — это тоже самое, что было в Bitcoin: аккаунт, единственной опцией которого является пересылка цифровой валюты (создание исходящей транзакции). В UNB путем создания и подписания транзакций с такого аккаунта можно пересылать сообщения. В случае контракт-аккаунта каждый раз, когда аккаунт получает входящее сообщение, активируется код смарт контракта, который может открывать сообщению возможность записи в хранилище и чтения оттуда информации, посылать другие сообщения и создавать другие смарт контракты. Словосочетание "смарт контракт" является короткой формой для "контракт-аккаунт".

Важным следствием механизма сообщений является принцип равноправия UNB-аккаунтов, управляемых контрактом и человеком — контракт-аккаунты имеют весь тот же функционал, что доступен аккаунтам, управляемым человеком, включая возможность пересылать сообщения и создавать другие контракты. Это приводит к тому, что контракты могут как люди, играть в сообществе разные роли одновременно.

В этом заключается важное преимущество смарт контрактов, позволяющее реализовывать на блокчейне совершенно недоступные ранее способы обработки информации и взаимодействия информационных систем, по-новому выстраивая бизнес-процессы.

1.3. Транзакции

Информация в блокчейне передается в виде сообщений - транзакций, которые записываются в блоки. Сообщение в UNB может быть создано как внешним источником (человеком, системой), так и смарт контрактом и содержит двоичные данные. Получатель UNB-сообщения, если это контракт-аккаунт, имеет возможность "ответить" автору на него.

Транзакция в блокчейне UNB является сообщением определенного формата (пакетом данных с цифровой подписью с применением закрытого ключа), передаваемого с определенного аккаунта на другой. Формат сообщения приведен в таблице 1.

Табл.1 Структура блокчейн транзакции

blockHash	хэш блока в котором записана транкация
blockNumber	номер блока в котором записана транкация
from	адрес отправителя
hash	хэш транзакции

input	данные, включенные в транзакцию
nonce	количество транзакций, посланных с адреса отправителя
to	адрес получателя
transactionIndex	индекс (номер позиции) транзакции в блоке
value	количество передаваемого актива
v	идентификатор восстановления цифровой подписи ECDSA (id)
r	компонент цифровой подписи ECDSA (r)
s	компонент цифровой подписи ECDSA (s)

Транзакции в блокчейне бывают двух типов: вызов сообщения и создание смарт контракта.

Транзакция вызова сообщения применяется для передачи сообщений от одной учетной записи к другой (обычно используется термин простая транзакция). Примеры - передача актива от одного аккаунта другому, вызов функции смарт контракта.

Также есть специальный тип транзакций и специальный тип сообщений для создания смарт контрактов. Транзакция создания смарт контракта имеет результатом создание нового смарт контракта (часто используется термины загрузка или деплой смарт контракта). Это означает, что когда данная транзакция выполнена успешно, она создает новый контракт-аккаунт со связанным с ним исполняемым кодом.

1.4. Цепочка блоков транзакций

Транзакция считается подтвержденной (т.е. завершенной и достоверной), когда проверены её формат и подписи, и когда сама транзакция объединена в группу с несколькими другими и записана в специальную структуру данных — блок. Проверку формата сообщения и валидность параметров транзакции осуществляет нода блокчейна. При отсутствии ошибки нода вычисляет хэш транзакции и она помещается в пул транзакций блокчейна для включения ее в очередной блок.

Структура блока блокчейна UNB приведена в таблице 2.

Табл.2 Структура блока блокчейна

Параметр	Описание
Заголовок блока	
number	Номер блока
hash	Хэш заголовка блока
parentHash	Хэш заголовка предыдущего (родительского) блока

nonce	Служебное поле
sha3Uncles	Хэш текущего списка блоков оммеров
logsBloom	Логи (информация для записи в журналы)
transactionsRoot	Хэш корневого узла префиксного дерева, содержащий все транзакции, которые перечислены в данном блоке
stateRoot	Хэш корневого узла состояния префиксного дерева
miner	Аккаунт валидатора
extraData	Служебное поле (дополнительные данные)
size	Размер блока
timestamp	Метка времени создания блока
Блок информации о транзакциях	
transaction	Массив с информацией о транзакциях

Валидаторы анализируют состояние пула транзакций и осуществляют формирование цепочки блоков по определенным правилам, которые составляют алгоритм консенсуса (см. далее). Физически цепочка блоков хранится в специальной базе данных на диске сервера (используется LevelDB).

Все блоки выстроены в одну цепочку, которая содержит информацию обо всех совершенных когда-либо операциях в блокчейне. Содержимое блоков защищено от изменений, так как каждый блок содержит информацию о предыдущем блоке.

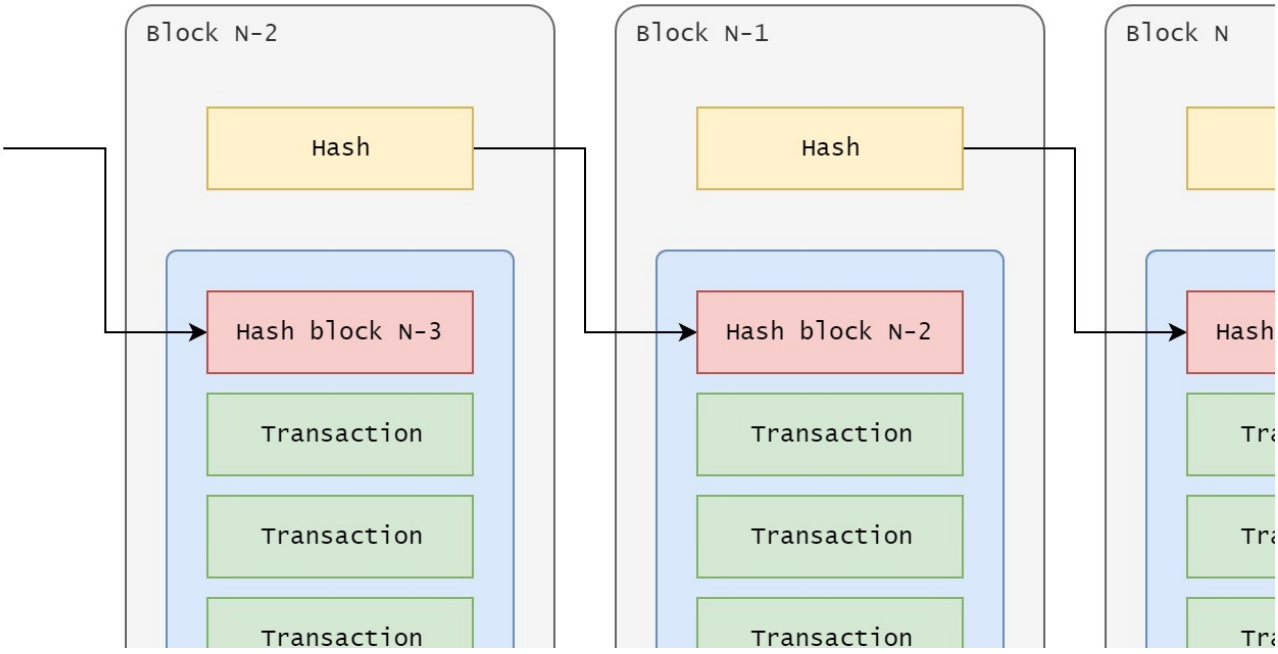

Рис.1. Блокчейн

1.5. Дерево Меркла

Блокчейн представляет собой единое хранилище состояний всех аккаунтов. Общее состояние всех аккаунтов хранится в специальной структуре данных – префиксном дереве Меркла.

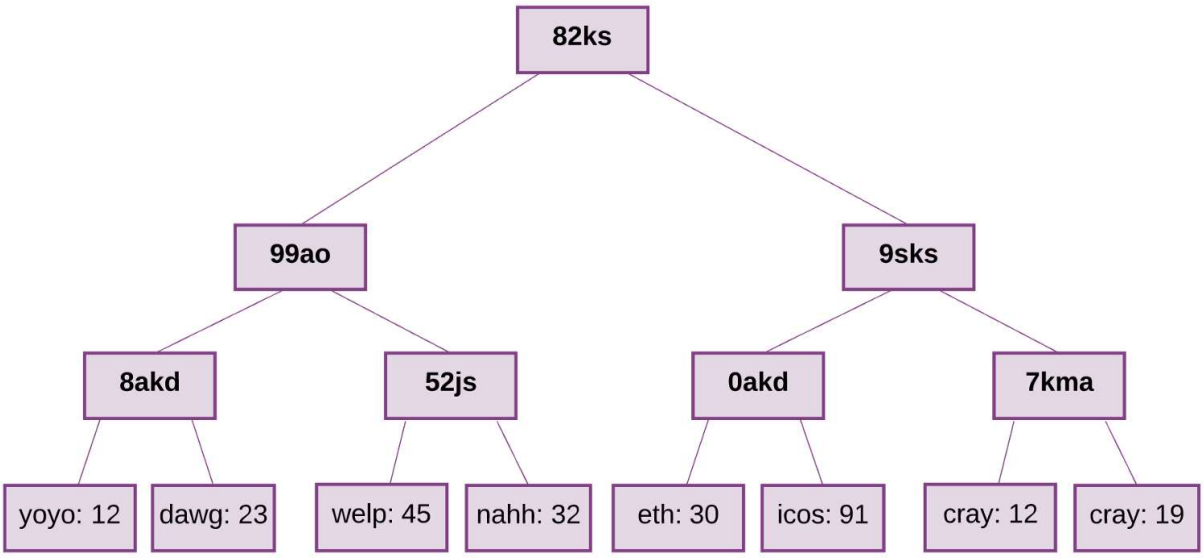

Рис. 2. Дерево Меркла

Дерево Меркла (Merkle trie) представляет собой тип двоичного файла, состоящего из набора узлов, которые включают:

- определенное количество конечных узлов, которые располагаются в нижней части дерева, содержащего базовые данные;
- набор промежуточных узлов, при этом каждый узел представляет собой хэш двух его дочерних узлов;
- один корневой узел, также образованный из хэша двух дочерних узлов, который представляет вершину дерева.

Данные, находящиеся в нижней части дерева, создаются путем разделения исходных данных на отдельные фрагменты, которые размещаются в ячейках хранения данных. После чего происходит их хэширование и процесс повторяется до тех пор, пока не получится один хэш (корневой хэш).

Подобная структура префиксного дерева также применяется для хранения транзакций.

Хэш в дереве Меркла распространяется от нижних ветвей к верхним, и если злоумышленник попытается заменить оригинальную транзакцию на поддельную в нижней части дерева Меркла, то это приведет к изменению хэша узла, а это, в свою очередь, приведет к изменению хэша располагающегося над ним узла и так до тех пор, пока, в конечном итоге, это не приведет к изменению корня.

Преимущество применения префиксного дерева Меркла заключается в том, что корневой узел данной структуры является криптографически зависимым от данных, хранящихся в дереве. Следовательно, хэш корневого узла может быть использован в качестве идентификатора безопасности для этих данных. Ввиду того, что в заголовок блока включен корневой хэш деревьев, а также их состояния, транзакции и информации, любой из узлов может проверять ту или иную часть состояния блокчейна UNB без

необходимости хранения всех состояний, которые могут быть потенциально неограниченными по размеру.

Возможность эффективного хранения информации в префиксном дереве Меркла является невероятно эффективным решением для так называемых тонких клиентов или тонких узлов блокчейна UNB.

1.6. Выполнение кода контракт-аккаунта

Для выполнения смарт контрактов в блокчейне UNB предусмотрена виртуальная вычислительная машина. Она полностью идентична Ethereum Virtual Machine (EVM), что обеспечивает совместимость смарт контрактов UNB и Ethereum. EVM является модулем блокчейн клиента UNB.

Код смарт контракта записывается в блокчейн UNB на низкоуровневом stack-based байткод-языке, который называется "код виртуальной машины Эфириум" или "EVM-код". Сам смарт контракт пишется на высокоуровневом языке Solidity и затем компилируется в EVM-код. Код состоит из последовательности байтов, каждый байт представляет какую-то операцию. В общем случае, выполнение кода — это цикл, состоящий из выполнения операции текущего байта и последующего увеличения номера обрабатываемого байта на 1. Этот процесс может остановится при достижении конца кода, ошибке в коде или после выполнения встретившейся в коде инструкции STOP/RETURN. Операции кода могут обращаться к трем типам памяти для размещения информации:

- стек, last-in-first-out контейнер, куда можно записывать и откуда можно извлекать значения;
- оперативная память, байтовый массив произвольного размера;
- хранилище контракта, информация в котором хранится парами "ключ-значение".

В хранилище параметры могут храниться долговременно, в отличие от стека и памяти, которые очищаются после выполнения кода.

Код смарт контракта имеет доступ к информации входящего сообщения, а также к информации в заголовке блока. Смарт контракт может вернуть данные (байтовый массив) на выходе.

Порядок выполнения EVM-кода следующий. При каждом раунде выполнения берется следующий байт EVM-кода и выполняется инструкция, соответствующая этому байту. Во время вычисления текущее состояние блокчейна содержится в наборе данных (block_state, transaction, message, code, memory, stack, pcs), где block_state — полное состояние, содержащее информацию о балансах и состояниях хранилищ всех аккаунтов. Каждая инструкция по-своему действует на этот набор данных. Для примера, ADD достает два элемента из стека и помещает в стэк их сумму. В целом набор инструкций EVM-кода соответствует стандартным двоичным кодам языка assembler.

Для упрощения разработки смарт контрактов сообществом программистов экосистемы Ethereum разработано несколько высокоуровневых языков программирования, компиляторов и инструментов для разработки, отладки и деплоя смарт контрактов.

В Системе применяются смарт контракты на языке Solidity. Компилятор смарт контрактов с языка Solidity включен в состав модулей блокчейн клиента UNB. Также в состав Системы включены сервисы компиляции и деплоя смарт-контрактов.

1.7. Логирование событий

В блокчейне UNB предусмотрена возможность вести журналы логов, цель которых записывать информацию о различных событиях, транзакциях и сообщениях.

Запись журнала включает;

- адрес учетной записи регистратора (аккаунт или контракт-аккаунт);
- ряд задач, которые отображают различные события, выполненные для текущей транзакции;
- данные, которые имеют отношение к данным событиям.

Для смарт контракта предусмотрена возможность прямого создания записи в таком журнале с помощью объявления события (Event), данные о котором требуется записать.

Записи журналов хранятся в особой структуре - фильтре Блума, благодаря чему обеспечивается возможность эффективно хранить практически бесконечное количество данных.

Фильтр Блума представляет собой битовый массив из m бит. Изначально, когда структура данных хранит пустое множество, все m бит обнулены. Пользователь должен определить k независимых хеш-функций h1, …, hk, отображающих каждый элемент в одну из m позиций битового массива достаточно равномерным образом. Для добавления элемента e необходимо записать единицы на каждую из позиций h1(e), …, hk(e) битового массива.

Для проверки принадлежности элемента e к множеству хранимых элементов, необходимо проверить состояние битов h1(e), …, hk(e). Если хотя бы один из них равен нулю, элемент не может принадлежать множеству (иначе бы при его добавлении все эти биты были установлены). Если все они равны единице, то структура данных сообщает, что e принадлежит множеству.

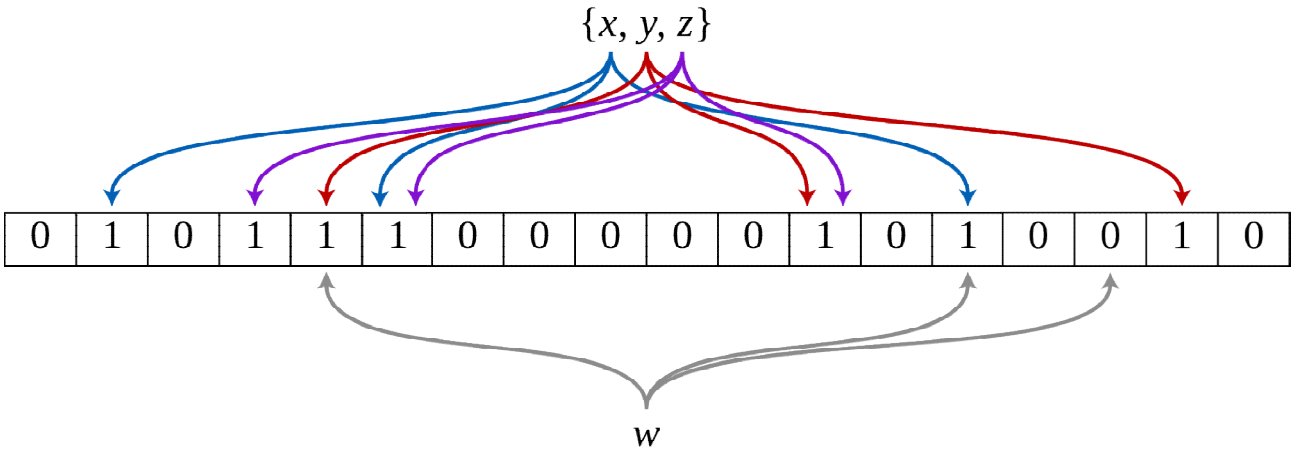

Рис. 3. Пример фильтра Блума.

На рисунке XXX приведен пример фильтра Блума с m=18 и k=3, хранящего множество {x, y, z}. Цветные стрелки указывают на места в битовом массиве, соответствующие каждому элементу множества. Этот фильтр Блума определил, что элемент w не входит в множество, так как один из соответствующих ему битов равен нулю.

Фильтр Блума используется для ускорения поиска в системе хранения ключ-значение. Значения хранятся на диске, который имеет медленное время доступа, решения фильтра Блума намного быстрее. Общая скорость ответа системы с фильтром Блума значительно выше, чем без фильтра Блума.

Ведение логов с использованием фильтра Блума реализуется блокчейн клиентом goUNB. Он также предоставляет программный интерфейс для доступа к логам.

2. Клиент блокчейна UNB

Блокчейн UNB представляет собой модифицированную версию блокчейна Ethereum. Для создания goUNB - клиента блокчейна UNB был взят за основу исходный код клиента geth Ethereum на языке Golang и внесены необходимые изменения. Реализация такого подхода возможна благодаря тому, что Ethereum является свободно-распространяемым программным обеспечением и широко используется при создании различных блокчейнов и децентрализованных систем по всему миру. При создании клиента goUNB были соблюдены все условия лицензии geth.

Модернизация клиента осуществлена с целью повышения его быстродействия в закрытых блокчейнах. В goUNB был разработан новый алгоритм консенсуса Kashi Dynamic Proof of Authority (DPOA). Его особенностью является динамическое управление временем формирования новых блоков. В отличие от исходного Clique POA (geth) в алгоритме Kashi DPOA (goUNB) время формирования блоков не фиксировано и определяется скоростью поступления транзакций. Оно может составлять от 50 ms до нескольких секунд, а при отсутствии транзакций новые блоки не формируются.

Другой особенностью блокчейна UNB является отсутствие комиссий при формировании транзакций. Это делает ненужным использование внутренней криптовалюты и так называемого “топлива” для выполнения транзакций и смарт контрактов (как Ether и gas в Ethereum). Это позволило при создании клиента goUNB исключить расчеты и проверки требуемого количества gas и его стоимости, что значительно увеличивает параметры быстродействия goUNB. Вопросы, связанные с безопасностью смарт контрактов и спама транзакций в закрытом блокчейне UNB решаются на уровне сервисов Системы, в том числе с помощью гибридного алгоритма консенсуса.

Другие принципы и методика формирования и записи транзакций, формирования блоков блокчейна остались без изменения, что обеспечивает полную совместимость блокчейнов UNB и Ethereum на уровне смарт контрактов.

3. Алгоритм консенсуса

3.1. Базовый алгоритм консенсуса

В блокчейне UNB используется алгоритм консенсуса Kashi Dynamic Proof of Authority (DPOA). Он основан на алгоритме Clique Proof of Authority, который применяется в блокчейне Ethereum.

Алгоритм PoA (Proof of Authority, дословно - доказательство полномочий) - это протокол, где только заранее определенные авторизованные валидаторы (в Ethereum они называются sealer, подписчик) могут создавать новые блоки. В этом отличие от PoW, где майнеры соревнуются в решении сложной задачи (поиск специального значения nonce при котором хэш заголовка блока начинается с последовательности определенного число нулей).

В PoA авторизованные валидаторы формируют и подписывают блоки по очереди. Очередность устанавливается по возрастанию адресов аккаунтов валидаторов в блокчейне (первый номер в очереди - валидатор с самым маленьким значением адреса, последний - с самым большим).

Для добавления нового блока в блокчейн необходимо, чтобы его подписали половина и более авторизованных валидаторов. Т.е. действует правило:

$$N_{подп} \geq N_{общ} /2 + 1.$$

Так, чтобы новый блок добавился в блокчейн при двух авторизованных валидаторах в сети, необходима подпись обоих, при трех - достаточно двух подписей, при четырех - трех, при пяти - трех и т.д.

Адреса аккаунтов авторизованных валидаторов в блокчейне записываются в заголовки блоков в специальное поле extraData. Для добавления нового или удаления существующего авторизованного валидатора в POA реализован механизм голосования, обеспечивающий децентрализацию управления блокчейном. Правом голосования обладают все авторизованные валидаторы. Для включения или исключения валидатора в список авторизованных необходимо чтобы за это решение проголосовали половина и более авторизованных валидаторов.

Первоначальная конфигурация блокчейна хранится в его первом блоке, который называется genesis блок. В POA в genesis блок дополнительно записывается информация об адресах аккаунтов авторизованных валидаторов, которые отвечают за формирование блокчейна на начальном этапе. В дальнейшем эти валидаторы, используя механизм голосования, могут изменить список авторизованных валидаторов.

Для управления списком авторизованных валидаторов используются специальные команды консоли клиента goUNB - kashi.proposal.

В Clique POA блоки формируются через определенный фиксированный интервал времени, который для совместимости с основной сетью Ethereum по умолчанию принят равным 15 секундам. В genesis блоке параметр period определяет интервал формирования блоков в секундах. Для увеличения быстродействия блокчейна в закрытых сетях рекомендуется уменьшать данный параметр. Однако при этом резко возрастает объем памяти, занимаемый блокчейном, поскольку блоки формируются постоянно, в том числе и при отсутствии транзакций. Самое высокое быстродействие блокчейна обеспечивается при установлении интервала 1 секунда (дробные значения параметра не допускаются). Однако даже при отсутствии транзакций объем блокчейна будет расти со скоростью примерно 2 мегабайта в час (в сутки - 48 MB, в год - более 17 GB “мусора” только на одном сервере, а в блокчейне их несколько) , что неприемлемо.

В экспериментальных целях в Cilque POA допускается устанавливать период формирования блоков равным 0. При этом при отсутствии транзакций новые блоки не формируются. Однако когда скорость поступления транзакций резко увеличивается, блоки начинают формироваться с максимально возможной скоростью. При этом формируются практически пустые (с единицами транзакций, а иногда и без транзакций) новые блоки, а их количество достигает нескольких десятков в секунду. Это значительно увеличивает объем данных блокчейна и снижает его быстродействие за счет большого числа операций записи данных на диск.

С целью увеличения быстродействия и оптимизации размера блокчейна Разработчиком реализован алгоритм Kashi Dynamic Proof of Authority. В отличие от Clique в новом алгоритме время формирования новых блоков задается с точностью до миллисекунд и зависит от скорости транзакций. Минимальный интервал формирования блоков составляет 100 миллисекунд при скорости транзакций 10000 TPS. При такой скорости транзакций в блокчейне UNB будет формироваться в среднем 10 новых блоков в секунду, содержащих по 1000 транзакций. Данные параметры приведены справочно, их оптимальные значения зависят от характеристик серверов и сетевой инфраструктуры, на которых функционирует блокчейн UNB и устанавливаются опытным путем.

3.2. Гибридный алгоритм консенсуса

Приведенный выше алгоритм консенсуса является базовым. Именно он определяет правила формирования блоков в блокчейне UNB. Одним из вариантов реализации DPOA могло бы быть использование смарт контракта для хранения информации об авторизованных валидаторах, которые имеют право формирования блоков.

Подобные подходы возможны и встречаются на практике. Однако существующий опыт показывает, что использование смарт контракта в этих целях сильно ограничивает быстродействие блокчейна. Причина этого заключается в том, что процесс формирования блоков не может быть произвольно остановлен при необходимости внести изменения в список авторизованных валидаторов. Таким образом, при синхронизации изменений остутствует доступ к синхронизированному состоянию, что приводит к непрогнозируемым задержкам.

Однако смарт контракт может с успехом быть применен для управления нодами, формирующими и транслирующими транзакции пользователей. Поскольку эти ноды также могут быть авторизованы как валидаторы, имеет смысл говорить о расширении алгоритма консенсуса.

Таким образом, гибридный алгоритм консенсуса UNB - это алгоритм управления доступностью нод, основанный на смарт контракте (контрактах) блокчейна UNB. Он является своеобразной надстройкой, дополнительным уровнем логики для базового алгоритма консенсуса Kashi DPOA и позволяет автоматизировать процесс управления нодами.

Возможность использования гибридного алгоритма консенсуса предусмотрена в данной версии клиента goUNB и может применяться без ограничений.

4. Топология блокчейна

4.1. Узлы блокчейна

На логическом уровне блокчейн UNB представляет собой совокупность логических узлов, называемых нодами, принадлежащих различным организациям, что обеспечивает децентрализацию. Ноды взаимодействуют между собой, обеспечивая передачу транзакций для включения в очередной блок, формирование и синхронизацию блоков.

Каждый логический узел, входящий в состав UNB, разворачивается на отдельном выделенном (виртуальном) сервере. Он содержит блокчейн клиент goUNB и различный набор сервисов (микросервисов). Конфигурация логического узла определяется администратором исходя из требований и возможностей конкретной организации. Архитектура типового узла блокчейна UNB приведена на Рисунке. 4.

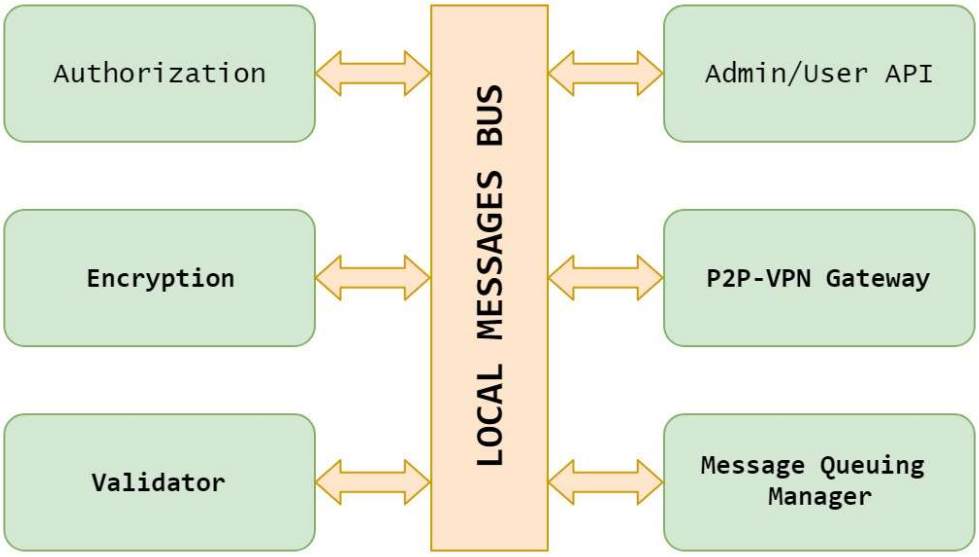

Рис. 4. Структура логического узла блокчейна

С целью повышения надежности и обеспечения масштабируемости блокчейна UNB начальный узел блокчейна (и, возможно, некоторые другие критически важные узлы) реализуются в виде кластера, состоящего минимум из двух мастер-нод и двух валидаторов, которые должны быть размещены на разных физических (желательно) и выделенных (обязательно) серверах. При этом мастер-ноды обеспечивают прием и обработку транзакций пользователей и передачу их в нужном формате валидаторам для включения в очередной блок блокчейна. Ноды в кластере работают поочередно, распределяя нагрузку. При этом мастер-нодами управляет балансировщик нагрузки, а валидаторы делают это автоматически благодаря алгоритму консенсуса. На Рисунке 5 приведена структура кластера нод узла блокчейна.

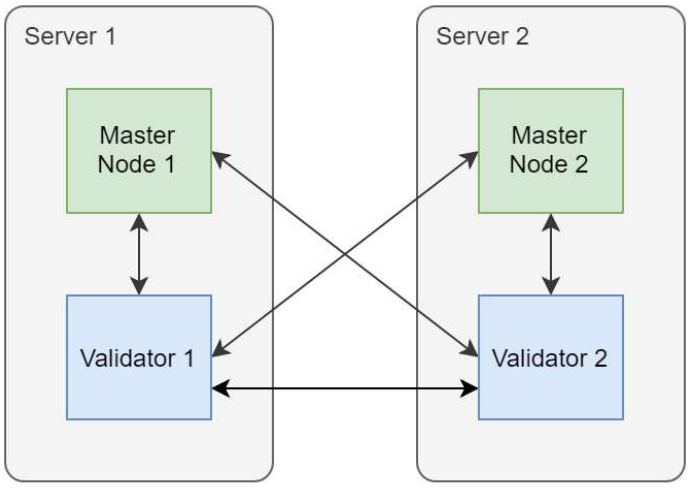

Рис. 5. Кластер нод узла блокчейна

Также с целью повышения быстродействия и обеспечения масштабируемости блокчейна требуется оптимизация количества нод и связей между ними (управление топологией блокчейна). При этом опытным путем определяются характеристики серверов и сетевой инфраструктуры, на которых развернут блокчейн UNB и устанавливаются предварительные параметры конфигурации нод. В дальнейшем используются инструменты нагрузочного тестирования и подбираются оптимальные параметры.

Пример оптимизированной топологии блокчейна UNB для трех организаций приведен на Рисунке 6.

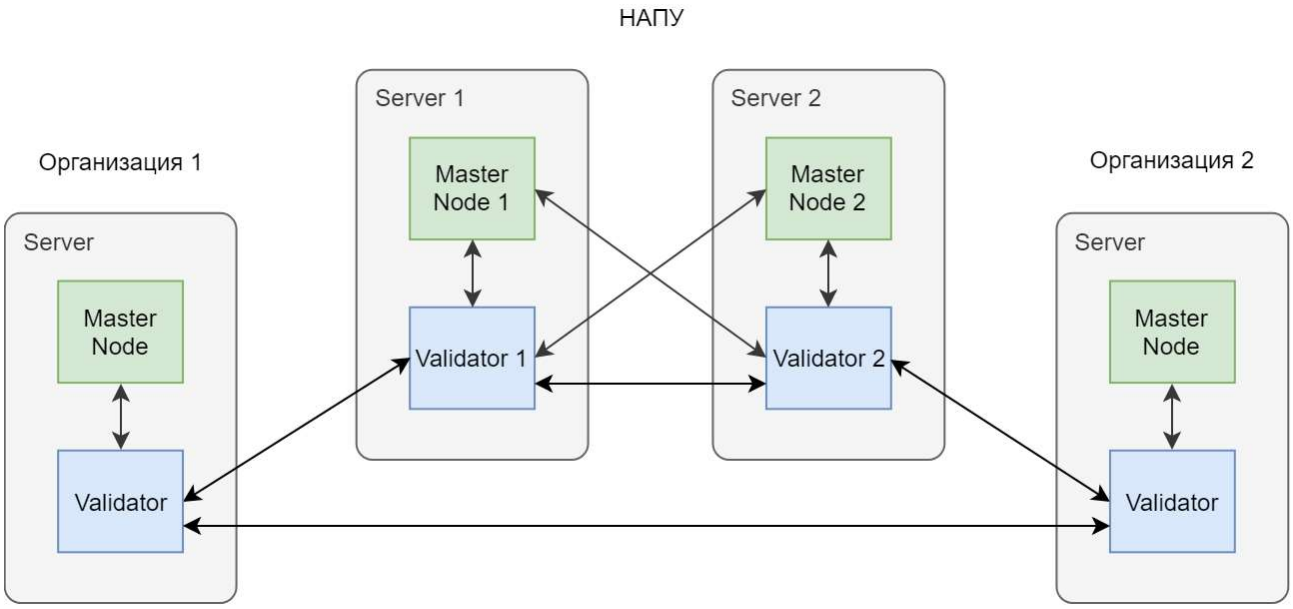

Рис. 6. Топология блокчейна UNB для трех организаций

4.2. Сайдчейны

Сайдчейн - это “дочерняя” цепочка блоков данных основного блокчейна, которая функционирует как самостоятельный блокчейн, аналогичный UNB. Сайдчейн предназначен для хранения информации ограниченного типа или определенной группы пользователей. Хэши заголовков блоков сайдчейна периодически записывается в основной блокчейн, чем обеспечивается дополнительная защита сайдчейна от несанкционированных изменений.

Применение сайдчейнов позволяет разгрузить основной блокчейн UNB, освободив его для записи критически важной информации,прежде всего информации о конфигурации Системы, системных смарт контрактах и др. Это решает проблему масштабирования блокчейна, повышает надежность и безопасность и увеличивает быстродействие Системы в целом. Данная технология масштабирования блокчейна по сути является шардингом и позволяет легко увеличивать быстродействие Системы в целом.

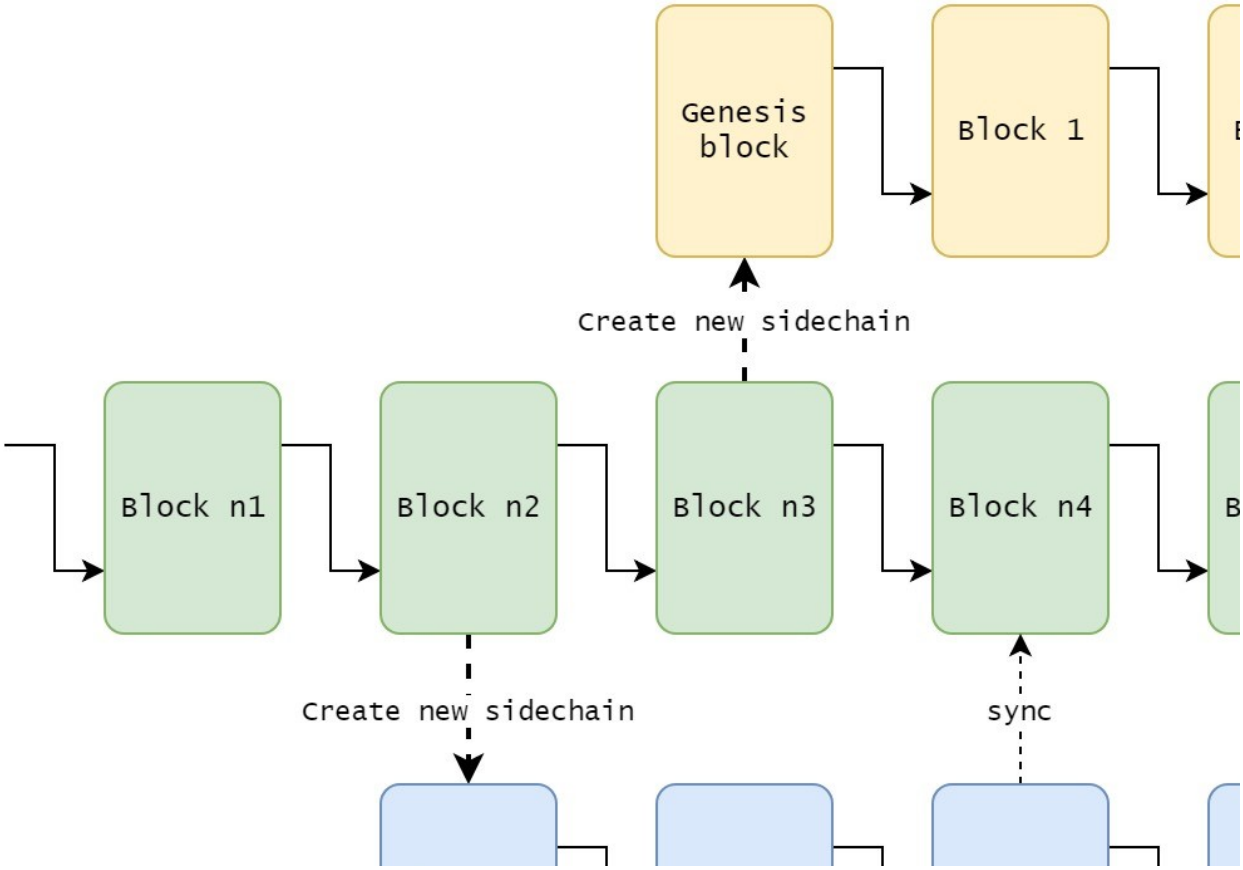

Рис.7. Схема Блокчейн - сайдчейны