ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ
«Программно-аппаратные средства технологий распределенного реестра и блокчейн»

Методические рекомендации ТК 159
МР 159.5.001 - 2020

Протокол обмена сообщениями между распределенными реестрами с децентрализованным подтверждением пересылаемых сообщений

Москва 2020

ПРЕДИСЛОВИЕ

1 РАЗРАБОТАНЫ коллективом экспертов Рабочей Группы № 5 Технического комитета по стандартизации ТК 159 «Программно-аппаратные средства технологий распределенного реестра и блокчейн»:

- Дружинин Илья Александрович, ilya.druzhinin@fintechru.org
- Колотов Александр Васильевич, a.kolotov@innopolis.ru
- Каламбет Петр Игоревич, peter@kalambet.dev

2 ВНЕСЕНЫ Техническим комитетом по стандартизации ТК 159 «Программно-аппаратные средства технологий распределенного реестра и блокчейн»

СОДЕРЖАНИЕ

Введение

Настоящие рекомендации предназначены для построения взаимодействия информационных систем на базе технологии распределенных реестров, используемый способ именуется экспертным сообществом «мостом».

Существует несколько вариантов реализации «мостов», описанных в опубликованном ранее документе [1]. В настоящих рекомендациях рассматривается вариант распределенной передачи сообщений между распределенными реестрами с децентрализованным подтверждением пересылаемых сообщений. Данный подход позволит повысить безопасность, достоверность передаваемых данных, а также защиту от атак цензурирования и единой точки отказа.

Текущий статус документа

Настоящие рекомендации подготовлены для публичного обсуждения и дополнения экспертами рынка блокчейн-технологий в рамках последовательности разработки и утверждения проекта национального стандарта. Обратную связь, предложения, замечания следует высылать руководителю рабочей группы №5 «Взаимодействие систем распределенного реестра» ТК 159 (Дружинин Илья Александрович,ilya.druzhinin@fintechru.org).

План дополнительных разделов настоящих рекомендаций приведен в Приложении А.

1 Область применения

Настоящие рекомендации описывают протокол обмена сообщениями между распределенными реестрами с подтверждением приема и передачи сообщений, благодаря которому подтверждение пересылаемого сообщения происходит автоматически с достижением консенсуса.

Для организации взаимодействия необходимо участие двух и более распределенных реестров, один из которых должен исполнять роль маршрутизирующего распределенного реестра (МРР), у всех остальных – роль корпоративных распределенных реестров (КРР). В сети могут принимать участие стороны, которые не связаны напрямую друг с другом в рамках жизненного цикла определенного бизнес-сценария.

Участниками протокола должны быть однозначно идентифицированные корпоративные сети, занесенные в реестр участников организующей взаимодействие МРР, и обладающие внутренними цифровыми активами (далее - ЦА), которые могут быть переданы за их пределы. Маршрутизирующий распределенный реестр может не обладать таким внутренним ЦА.

Данный документ предназначен для архитекторов и разработчиков информационных систем на базе распределенного реестра, а также для сотрудников информационной безопасности для проверки соответствия требованиям настоящих

методических рекомендаций разрабатываемой информационной системы и функций взаимодействия со сторонними распределенным реестрами.

2. Нормативные ссылки

В настоящих рекомендациях нормативные ссылки не используются.

3 Термины, определения и обозначения

Термины

Византийская устойчивость (BFT) – свойство устойчивости протокола консенсуса к недобросовестному, вредоносному поведению.

Внереестровое программное обеспечение – программное обеспечение, действующие вне распределенного реестра и взаимодействующее с распределенном реестром посредством API распределенного реестра.

Консенсус – способ согласования состояния в распределенной среде.

Мост – протокол (архитектурное решение), объединяющий два и более распределенных реестра с использованием автономных агентов («оракулов»), для приема, проверки и передачи информации между распределенными реестрами [на основе ГОСТ 34.321-96 (пункт 2.39)].

Оракул – автономный агент, который находит и подтверждает события в распределенном реестре и передает эти данные в распределенный реестр для использования смарт-контрактами. Агенты могут образовывать сеть подтверждения для проверки достоверности события распределенного реестра.

Цифровой актив – информация, размещенная в информационной системе и представляющая ценность для ее участников.

Пользователь приложения – пользователь, который вызывает MPP API или смарт-контракт. Вызов API инициирует отправку транзакции в распределенный реестр.

Обозначения

ПСмК - прикладной смарт-контракт
СмК - смарт-контракт
ССмК - системный смарт-контракт
КРР - корпоративный распределенный реестр
МРР - маршрутизирующий распределенный реестр
ЦА - цифровой актив
API - Application Programming Interface

4 Общие положения

4.1. Описание

На техническом уровне «мост» представляет собой группу равноправных узлов, каждый из которых состоит из программно-аппаратного комплекса, включающего:

- узел подключаемого корпоративного распределенного реестра (**КРР**);
- узел маршрутизирующего распределенного реестра (**МРР**);
- набор системных смарт-контрактов (размещаемых в распределенных реестрах **МРР** и **КРР**);
- внереестровое программное обеспечение (далее – оракул) для взаимодействия со смарт-контрактами в распределенных реестрах.

Каждый узел «моста» физически должен быть участником обеих сетей (МРР, КРР).

4.2 Распределенные реестры и требования к ним

4.2.1 Маршрутизирующий распределенный реестр

Маршрутизирующий распределенный реестр должен, как правило, состоять из одноранговых узлов, список которых размещен в белом списке узлов маршрутизирующей одноранговой сети. А также обладать:

- механизмом согласования состояния в распределенном среде с византийской устойчивостью;
- смарт-контрактами для выполнения системных функций моста на уровне распределенного реестра (смарт-контрактов в виртуальной машине, исполнение программных функций в среде исполнения узла распределенного реестра);
- изолированностью логики и среды выполнения системных функций моста (в смарт-контрактах распределенного реестра) от внешней среды;
- смарт-контракты моста должны уметь сохранять свое состояние;
- доступом к событиям и состояниям СмК моста посредством API;
- МРР должны формировать одноранговые узлы без выделения отдельных участников;
- окончательной завершенностью истории изменения состояния моста;
- МРР должен поддерживать большое количество участников (>100).

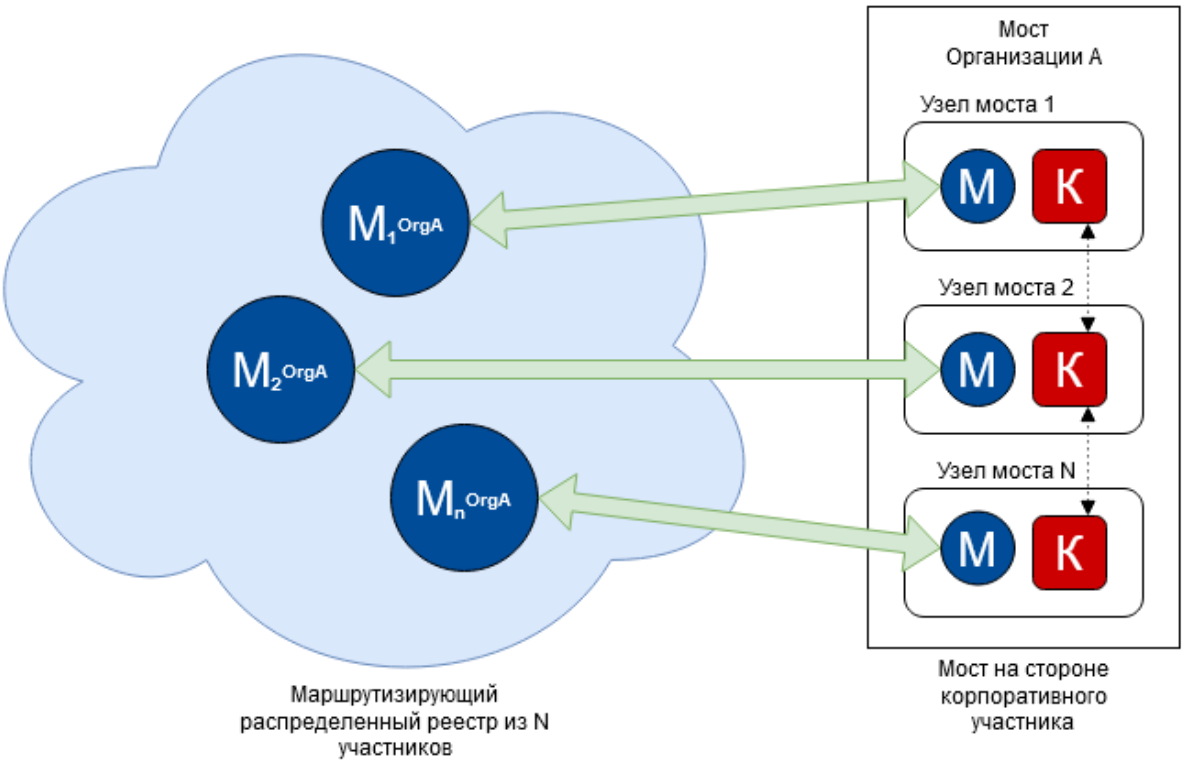
4.2.2 Корпоративный распределенный реестр

Корпоративный распределенный реестр – это распределенный реестр, принадлежащий конкретной компании, группе компаний, консорциуму и отвечающий бизнес-потребностям, ради которых он был разработан и введен в эксплуатацию. КРР может обладать внутренней ценностью (цифровыми активами), которые он может передавать в сторонние КРР. Для этого платформа, реализующая КРР, должна включать:

- окончательность и неподдельность транзакций;
- поддержку и возможность проверки подписи оракула(-ов);
- системные смарт-контракты для проверки, создания подписей;
- КРР должен реализовать общий для всех КРР API.

КРР API – это минимально необходимый набор функциональных возможностей платформы ТРР, на которой реализован узел КРР, для возможности взаимодействия с МРР. В состав возможностей должно включаться формирование мультиподписи для формирования и проверки подписи оракула. КРР API может реализовываться на смарт-контрактах.

4.3 Требования к мосту и принципы организации моста



Фигура 1

4.3.1 Устройство моста на стороне отдельного участника

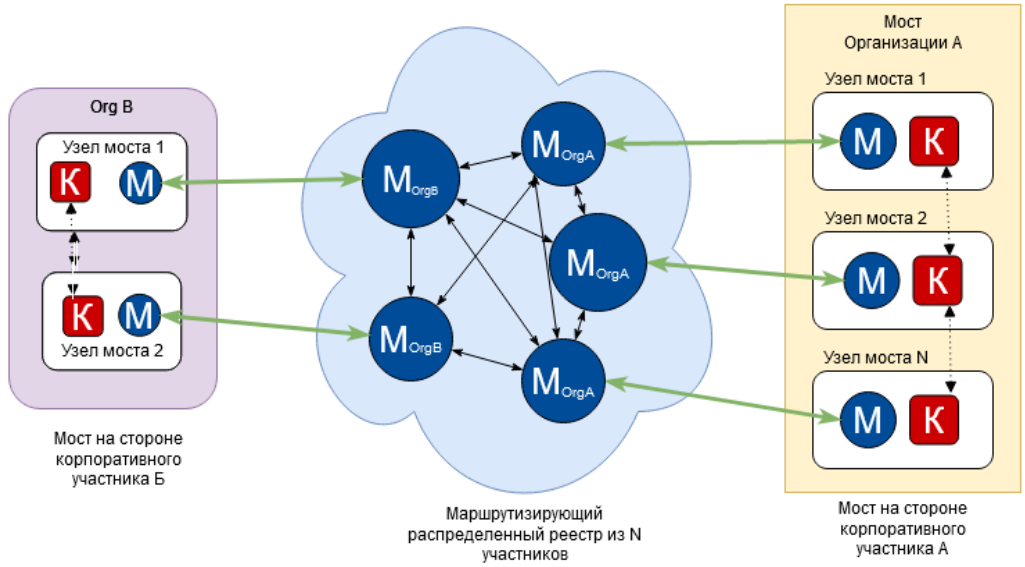
Мост на стороне определенного участника представляет собой группу узлов (от 1 до n), каждый из которых включает узел маршрутизирующего распределенного реестра и корпоративного распределенного реестра. При этом:

- узел маршрутизирующего распределенного реестра отвечает за прием внешних сообщений (на картинке - М);
- узел корпоративного распределенного реестра (на картинке - К) отвечает за подтверждение и обработку входящих сообщений и отправку сообщений в маршрутизирующий распределенный реестр;

- тождественность приема и получения сообщений достигается на основе голосования оракулов по каждому сообщению, если того требует конфигурация моста.

4.3.2 Устройство межорганизационного моста

Взаимодействие нескольких организаций с помощью моста реализуется без изменения предложенной схемы моста на стороне отдельного участника. Каждый мост подключается к маршрутизирующему распределенному реестру по общему принципу, при соответствии требованиям к корпоративным распределенным реестрам.



Фигура 2.

4.4 Смарт-контракты и требования к ним

Смарт-контракты моста делятся на прикладные и системные.

- Прикладные смарт-контракты (ПСмК) описывают бизнес логику, при которой в процессе своего жизненного цикла они отправляют сообщения в другие распределенные реестры (посредством вызова функций системных смарт-контрактов).
- Системный смарт-контракт (ССмК) отвечает за реализацию функций моста. Унифицированные для всех взаимодействующих КРР смарт-контракты предназначены для приема сообщений из прикладных смарт-контрактов, взаимодействия с оракулом и передачи сообщений в прикладной смарт-контракт КРР.

4.4.1 Прикладные смарт-контракты

Прикладные смарт-контракты отвечают за правила формирования исходящих сообщений и правила обработки входящих сообщений. В том числе ПСмК может:

- определять правила упорядочивания сообщений (абстракция очереди сообщений) и создания очередей входящих и исходящих сообщений, если это важно;
- фиксировать факт отправки сообщения в ССмК;

- проверять отправителя и адреса назначения (сигнатуры смарт-контракта, метода смарт-контракта) сообщения в списках участников сети МРР и/или КРР, чтобы избежать ошибочного получения или ложного сообщения атакующего;
- определять метод для приема входящих сообщений.

Прикладной смарт-контракт должен иметь уникальный адрес в КРР.

4.4.2 Системные смарт-контакты

Логически системные смарт-контракты описывают схему передачи и получения сообщений из одной сети в другую посредством организации исходящей (outbox) и входящей (inbox) очередей сообщений. Повышение доверия к схеме и способу передачи достигается за счет голосования за каждое из передаваемых и получаемых сообщений группой узлов, входящих в мост, если это предусмотрено конфигурацией.

ССмК отвечают за прием входящего от ПСмК КРР сообщения и отправку его в ССмК МРР, при этом:

- ССмК КРР должны иметь ограничения по списку ПСмК и пользователей, которые могут их вызывать;
- ССмК КРР может регистрировать получателей сообщений (ПСмК) из МРР для корректной адресации;
- ССмК может поддерживать блоклист смарт-контрактов, которые не могут взаимодействовать с ССмК;
- ССмК должен ставить в однозначное соответствие вызываемой сущности (ПСмК может регистрировать свой адрес в ССмК) уникальный/конкретный/ стандартизированный адрес в КРР.

Одним из способов оптимизации взаимодействия может быть совмещtнная система передачи и голосований за сообщения. Таким образом узлы моста могут осуществлять:

- раздельную передачу и голосование за сообщения – обосновано для корпоративных систем, в которых разные уровни ответственности у разных компонент, в целях обеспечения требований ИБ;
- совместную передачу и голосование за сообщение – проще для реализации, менее доверенное с точки зрения ИБ.

4.5 Оракул и требования к нему

Оракул – внереестровое (off-ledger, off-chain) программное обеспечение. Призван осуществлять взаимодействие между клиентами распределенного реестра (МРР, КРР) и смарт-контрактами, размещенными в реестрах.

Задачи оракула:

- подписываться на события и отслеживать события, создаваемые в ССмК (в МРР, КРР);
- вызывать методы ССмК в КРР;
- проверять целостность передаваемых сообщений;

- подписывать входящее сообщение в MPP при принятии(например, m-из-n подписей или сборка подписи);
- проверять сертификаты отправителя, прав участия (если реализована модель участия в MPP) (опционально);
- проверять правила финальности для MPP и KPP (организаторы MPP определяют точку финальности и правила финальности (технологическая, организационная).

4.6 Организационный уровень и требования к нему

4.6.1 Правила добавления участника

В смарт-контракт реестра участников MPP по определенным правилам записывается информация по новому добавленному KPP (количество узлов на стороне участника, адреса узлов в MPP).

4.6.2 Правила исключения участника

В смарт-контракте реестра участников MPP по определенным правилам (голосование участников, экономическая модель управления, судебное предписание) обновляется статус исключаемой KPP: адреса KPP вносятся в стоп-лист.

4.6.3 Развертывание, модификация ССмК

На этапе развертывания ССмК моста:

- определяется порог голосования за сообщение (зависит от реализации подписи сообщений оракулами);
- определяется список (в т.ч. список из одного) публичных ключей оракулов, которые передают сообщения;
- определяются технические ограничения на передаваемые сообщения.

4.6.4 Развертывание ПСмК

На этапе развертывания ПСмК:

- указывают адрес ССмК для отправки сообщения;
- определяют параметры, свойственные для принимающей стороны.

4.6.5 Модификация, обновление ПСмК

При обновлении с созданием нового экземпляра ПСмК указывают адрес предыдущего ПСмК, который содержит информацию, и создают функции запроса информации из предыдущего ПСмК.
При обновлении ПСмК без создания нового экземпляра ПСмК интерфейс взаимодействия с ССмК может оставаться прежним.
При обновлении ПСмК, если адрес ПСмК меняется, информацию по нового уникальному адресу следует передать в ССмК либо сохранить в специальном поле состояния ПСмК.

4.6.6 Прекращение деятельности ПСмК

При прекращении деятельности ПСмК должен возвращать вызывающей стороне информацию по статусу ПСмК.

4.6.7 Развертывание оракула

При развертывании оракула:

- задаются адреса ССмК МРР и ССмК КРР;
- задаются правила финальности, свойственные для КРР и МРР;
- задаются правила подписания сообщений (мультиподпись, сборка подписи);
- задаются проверки отправителя и адресата (опционально).

5 Протокол передачи сообщения

5.1 Протокол передачи сообщения

1. Инициатор сообщения (клиент или ПСмК КРР) отправляет закодированное сообщение в ССмК КРР, который:
    a. фиксирует факт отправки сообщения ПСмК;
    b. формирует голову сообщения на основе сообщения ПСмК;
    c. регистрирует (создает/генерирует) событие.
2. Оракул на стороне отправителя отслеживает события ССмК:
    a. этап проверки сообщения: оракул проверяет отправителя, финальность записи сообщения в КРР;
    b. этап отправки сообщения: формирует, подписывает и отправляет транзакцию с сообщение в МРР.
3. Произвольный узел моста прослушивает МРР на наличие сообщений для своего КРР. При получении сообщения производит:
    a. проверку финальности согласно правилам сообщения в МРР;
    b. подпись транзакции и отправляет сообщение в КРР.
4. ССмК КРР, *отвечающий за приём сообщений*, по факту получения сообщения от оракула, проверяет:
    a. зарегистрировано ли событие в ССмК;
    b. если событие не зарегистрировано – регистрирует сообщение, выставляя переданный голос в качестве начального значения, увеличивается счетчик;
    c. если событие уже зарегистрировано – добавляет переданный голос к списку уже добавленных (инкрементируется счетчик).
    d. контролирует, что каждый из участников может проголосовать всего один раз;
    e. проверяет, удовлетворяет ли требование порогу, установленному в правилах работы голосования ССмК;
    f. если удовлетворяется пороговому значению – отправляет сообщение ПСмК(-ам);
    g. если не удовлетворяет – ССмК ждет подтверждение от необходимого количества оракулов.

5.2 Протокол запроса информации из внешнего РР

1. Инициатор сообщения (клиент или ПСмК КРР) отправляет в ССмК закодированное сообщение с запросом на чтение, с информацией, что читаем, и указанием информации о получателе результата, который:
    a. фиксирует факт отправки сообщения ПСМК;
    b. формирует голову сообщения (тип сообщения (чтение или исполнение), что запрашиваем и кому направить ответ) на основе сообщения ПСмК;
    c. регистрирует событие.
2. Оракул на стороне отправителя отслеживает события ССмК КРР:

    a. этап проверки сообщения: оракул проверяет отправителя и получателя, финальность записи сообщения в КРР;
    b. этап отправки сообщения: формирует, подписывает и отправляет транзакцию с сообщение в МРР.
3. Произвольный узел моста прослушивает МРР на наличие сообщений для своего КРР. При получении сообщения производит:
    a. проверку финальности сообщения согласно правилам, установленным в МРР;
    b. оракул по временной отметке блока, в котором было сообщение в МРР, определяет номер блока КРР, в котором будет происходить чтение;
    c. направляет запрос на чтение по конкретному блоку КРР.
4. ССмК КРР, *отвечающий за прием сообщений*, по факту получения сообщения от оракула:
    a. отправляет сообщение ПСмК (-ам);
    b. при получении сообщения от ПСмК формирует сообщение с получателем;
    c. информирует оракула о наличии сообщения для получателя.
5. Оракул получает ответ от ССмК:
    a. этап проверки сообщения: оракул проверяет отправителя, финальность записи сообщения в КРР;
    b. этап отправки сообщения: формирует, подписывает и отправляет транзакцию с сообщение в МРР.

6 Состав сообщения

Сообщение представляет собой составной пакет из «головы» и «тела». Обе части сообщения являются набором байт, которые имеют стандартное расположение относительно друг друга (схема). Семантика сообщения не входит в логику работы моста, чтобы сделать такой подход максимально универсальным, и может быть реализована набором смарт-контрактов, описывающих требуемую бизнес-логику в каждой сети соответственно.

Размер заголовка сообщения имеет фиксированное значение, содержит служебную информацию для оракулов и служит для адресации основного сообщения.

Размер «тела» сообщения может меняться и определяется в зависимости от типа сообщения, нужд и особенностей сценария, к которому они применяются. Содержит основную, значимую информацию сообщения.

6.1 Поля сообщения

6.1.1 Заголовок сообщения

- version – версия используемого протокола
- message_id – уникальное значение сообщения в MPP, правила формирования определяются на этапе выбора или разработки платформы MPP
- source_chain_id – адрес КРР отправителя, указывается в реестре участников MPP, формируется по правилам MPP
- destination_chain_id – адрес КРР получателя, указывается в реестре участников MPP, формируется по правилам MPP
- sender_address – адрес ПСмК в КРР, который отправляет сообщение
- executor_address – удаленный ПСмК в КРР (destination_chain_id), который будет получать и выполнять сообщение
- datatype – указывается тип сообщения, write - для отправки сообщения в удаленный ПСмК, callback - для запроса информации в удаленном ПСмК

6.1.2 Тело сообщения

data (полезная нагрузка) – поле, которое содержит значимую информацию по пересылаемому сообщению

- method – вызываемый метод в удаленном ПСмК КРР
- params – произвольная длина, набор параметров такого метода, в том числе nonce для прикладных ПСМК, которым важно упорядочение сообщений
    - nonce – как основной инструмент определения очередности сообщений должен контролироваться кодом ПСмК, быть частью хранимого состояния ПСмК, отвечающего за приём сообщений в список исходящих

Ссылки

1. ИССЛЕДОВАНИЕ ПРОТОКОЛОВ ВЗАИМОДЕЙСТВИЯ РАСПРЕДЕЛЕННЫХ РЕЕСТРОВ

Приложение А. Развитие дополнительных разделов МР

1. Раздел «Типовая реализация». Описание реализации протокола в терминах выбранных платформ распределенных реестров (Мастерчейн, Hyperledger Fabric).
2. Раздел «Требования информационной безопасности». Описание целей безопасности, модели угроз и нарушителя, потенциальных атак на протокол и взаимодействующие ИС.
3. Раздел «Сценарии использования и бизнес-применение». Описание бизнес-сценариев, реализованных на протоколе.
4. Раздел «Формальная спецификация». Состав формальной спецификации и подходы к верификации протокола.
5. Приложение «Рекомендации по организации и эксплуатации МРР». Описание последовательности организационных шагов при запуске и эксплуатации МРР.