ФАБРИК
БЛОКЧЕЙН ПЛАТФОРМА

Руководство администратора

Блокчейн платформа “Фабрик”

Содержание:

1. Юридические уведомления
    1.1. Гарантия
    1.2. Авторские права
    1.3. Товарные знаки
2. Архитектура Блокчейн платформы “Фабрик”
    2.1. Что такое сеть блокчейн?
    2.2. Функциональные особенности Блокчейн платформы “Фабрик”
    2.3. Поддержка ГОСТ-алгоритмов криптографии
3. Процесс создания сети на Блокчейн платформе “Фабрик”
    3.1. Создание сети
    3.2. Центры сертификации
    3.3. Добавление сетевых администраторов
    3.4. Создание Консорциума
    3.5. Создание канала для Консорциума
    3.6. Узлы (пиры) и Реестры (леджеры)
    3.7. Приложения и чейнкод
    3.8. Завершение настройки сети
    3.9. Резюме сети
4. Глоссарий
5. Системные требования для установки демонстрационной сети
    5.1. Аппаратное обеспечение
    5.2. Операционные системы
    5.3. Безопасность
6. Подготовка системы к установке
    6.1. Установка пакетов Docker и Docker-compose
    6.2. Распаковка докер-образов
6.3. Конфигурация сети
6.4. Запуск сети
6.5. Остановка сети
7. Устранение неполадок

1. Юридические уведомления

1.1. Гарантия

Правообладатели не предоставляют никаких гарантий в отношении этого документа, включая, в частности, подразумеваемые гарантии товарности и пригодности для определенной цели потребителя. Правообладатели не несут ответственности за неисправности, содержащиеся в настоящем документе, а также прямые, косвенные, случайные или специальные убытки в связи с предоставлением, эксплуатацией или использованием данного материала.

1.2. Авторские права

1.3. Товарные знаки

Docker является зарегистрированным товарным знаком корпорации Docker.
UNIX является зарегистрированным товарным знаком корпорации Open Group.

Поскольку мы заявляем, что используем товарные знаки исключительно в целях редакции и соответствуя интересам владельца товарного знака, мы не будем размещать символ товарного знака в каждом случае его использования.

2. Архитектура Блокчейн платформы “Фабрик”

2.1. Что такое сеть блокчейн?

Сеть блокчейн - это техническая инфраструктура, которая предоставляет приложениям сервисы распределенного реестра и смарт-контракты (чейнкод). В первую очередь, смарт-контракты используются для генерации транзакций. Транзакции впоследствии распределяются каждому участнику сети (ноду), где они записывают в копии распределенного реестра (леджера) и впоследствии являются неизменными. Пользователями приложений могут быть как конечные пользователи, использующие клиентские приложения, так и администраторы сети распределенных реестров.

2.2. Функциональные особенности Блокчейн платформы “Фабрик”

Главное отличие Блокчейн платформы “Фабрик” от других распределенных реестров заключается в том, что это первая действительно масштабируемая блокчейн-система для запуска распределенных приложений, которая поддерживает алгоритмы шифрования в соответствии со стандартами ГОСТ. Помимо этого она реализует модель с эксклюзивным доступом к обработке транзакций (англ. permissioned). Вместо открытой

системы без прав доступа, которая позволяет участвовать в сети без идентификации, члены сети Блокчейн платформы “Фабрик” регистрируются через доверенного поставщика услуг, согласно внутренней политике, одобренной и принятой между организациями сети.

Также Блокчейн платформа "Фабрик" позволяет не только создавать организации, но и объединять несколько организаций в консорциум для формирования сети, при этом уровни доступа организаций определяются набором политик, согласованных консорциумом при первоначальной настройке сети. В архитектуре платформы заложена возможность в процессе эволюции сети менять политику в зависимости от задач возлагаемых на сеть ее участниками.

Блокчейн платформа “Фабрик” также предлагает возможность создавать каналы, позволяя группе участников создавать отдельный изолированный реестр (леджер) транзакций. Эта особенность заложена в архитектуру платформы на те случаи, когда участники сети могут быть конкурентами или не хотят, чтобы какие-либо данные, передаваемые по сети, были известны другим участникам сети без санкции на это со стороны инициатора данных. Если два участника образуют канал, то только эти участники (любая утечка данных не возможна) имеют копию распределенного реестра (леджера) для этого канала.

2.3. Поддержка ГОСТ-алгоритмов криптографии

Платформа Блокчейн платформы "Фабрик" выполняет все операции с применением криптографических стандартов Российской Федерации ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012. Операции по шифрованию данных, проверке целостности и управлению сертификатами выполняются посредством сертифицированных криптопровайдеров (VipNet CSP).

ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» — российский криптографический стандарт, описывающий алгоритмы формирования и проверки электронной подписи, принятый и введенный в действие вместо ГОСТ Р 34.10-2001 Приказом Федерального агентства по техническому регулированию и метрологии от 7 августа 2012 года №215-ст.

ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хеширования» — действующий российский криптографический стандарт, определяющий алгоритм и процедуру вычисления хеш-функции. Разработан Центром защиты информации и специальной связи ФСБ России с участием ОАО «ИнфоТеКС» и введен в действие 1 января 2013 года.

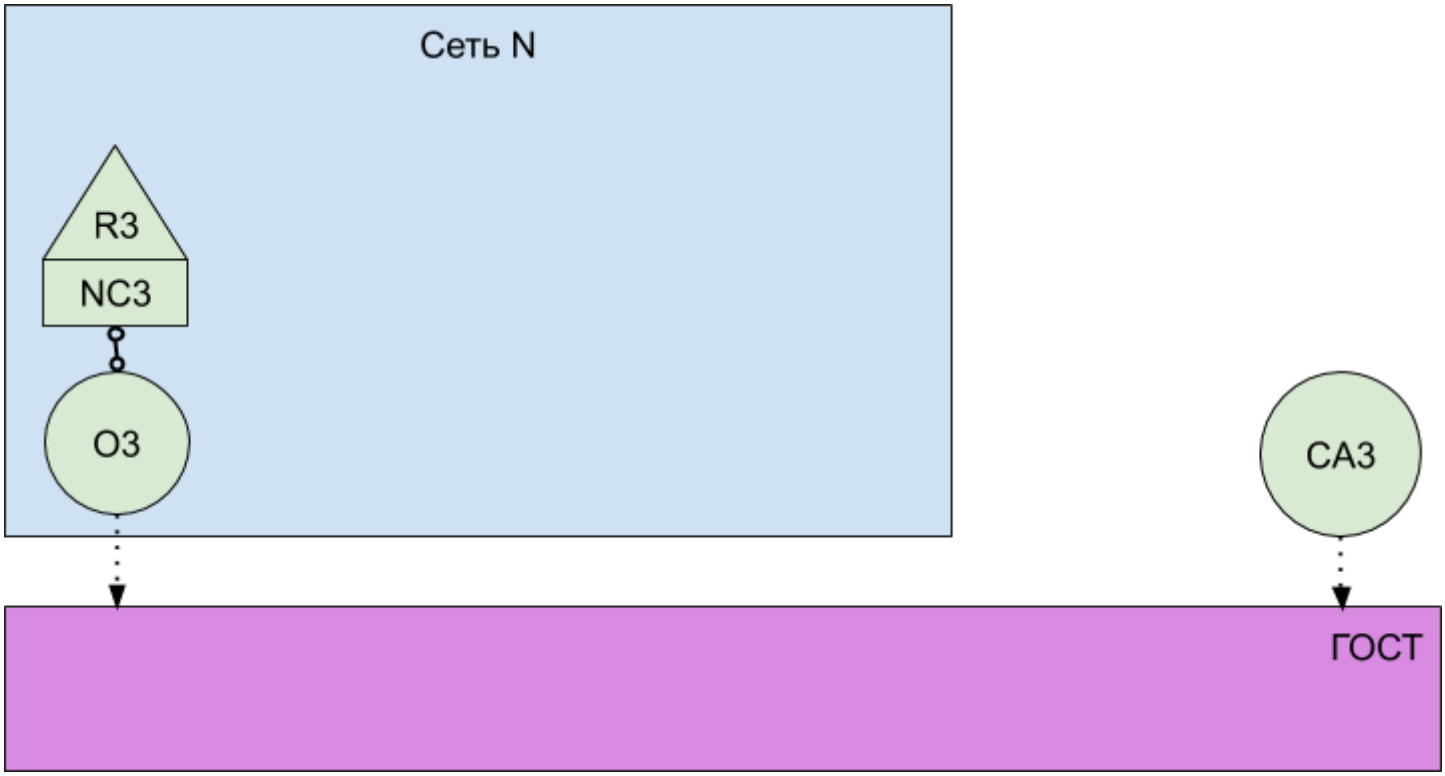
Использование криптографии на Блокчейн платформе "Фабрик" проверяется на следующих уровнях:

1. Вызов функций криптопровайдеров
2. Соответствие криптографических материалов администраторов организаций и каналов ГОСТ-34.10/34.11
3. Соответствие криптографических материалов пользователей ГОСТ-34.10/34.11
4. Использование крипто-материалов используемых во время создания каналов
5. Анализ криптоматерилов, сохраняемых в двоичных данных в распределенном реестре на предмет соответствия ГОСТ-ам и соответствия созданным ранее крипто-материалам

3. Процесс создания сети на Блокчейн платформе "Фабрик"

3.1. Создание сети

Создания основы для сети:



Сеть формируется при запуске Службы формирования новых блоков-сервера (Ordering Service). В нашем примере в сети N служба Ordering Service, состоящая из одного узла (ордерера) O3, настроена в соответствии с сетевой конфигурацией NC3, которая предоставляет административные права организации R3. На сетевом уровне центр сертификации CA3 используется для выдачи удостоверений администраторам и сетевым узлам организации R3. CA3 и O3 используют обращение к серверу для

формирования и проверки квалифицированных сертификатов в соответствии со стандартом ГОСТ.

Первое, что определяет сеть N - это служба Ordering Service, состоящая из узла O3. Для более простого восприятия можно рассматривать эту службу как начальную точку администрирования сети. Ордерер O3 (узел службы формирования новых блоков) изначально настраивается и запускается администратором, и размещается в организации R3. Конфигурация NC3 содержит политики, которые описывают начальный набор административных возможностей для сети.

3.2. Центры сертификации

Центр сертификации CA используется для выдачи сертификатов администраторам и сетевым узлам. CA играет ключевую роль в сети, поскольку он выдает сертификаты, которые могут использоваться для идентификации компонентов сети как принадлежащих организации R3. Сертификаты, выданные центрами сертификации, также можно использовать для подписания транзакций: указание того, что организация подтверждает результат транзакции - это предварительное условие принятия транзакции в реестр (леджер).

Различные компоненты Блокчейн платформы “Фабрик” используют сертификаты, чтобы идентифицировать себя или указать принадлежность к конкретной организации. Для этого обычно существует несколько центров сертификации, поддерживающих сеть распределенных реестров - то есть разные организации используют разные CA.

Служба, через которую участники осуществляют проверку принадлежности объекта к той или иной организации или каналу, называется Membership Services. Конфигурация сети NC3 использует эту службу для идентификации свойств сертификатов выданных CA, которые связывают владельцев сертификатов с организацией R3. Затем сетевая конфигурация NC3 использует Membership Services в политиках для предоставления субъектам из R3 определенных прав на сетевые ресурсы. Примером такой политики является определение администраторов в организации R3, которые могут добавлять новые организации в сеть.

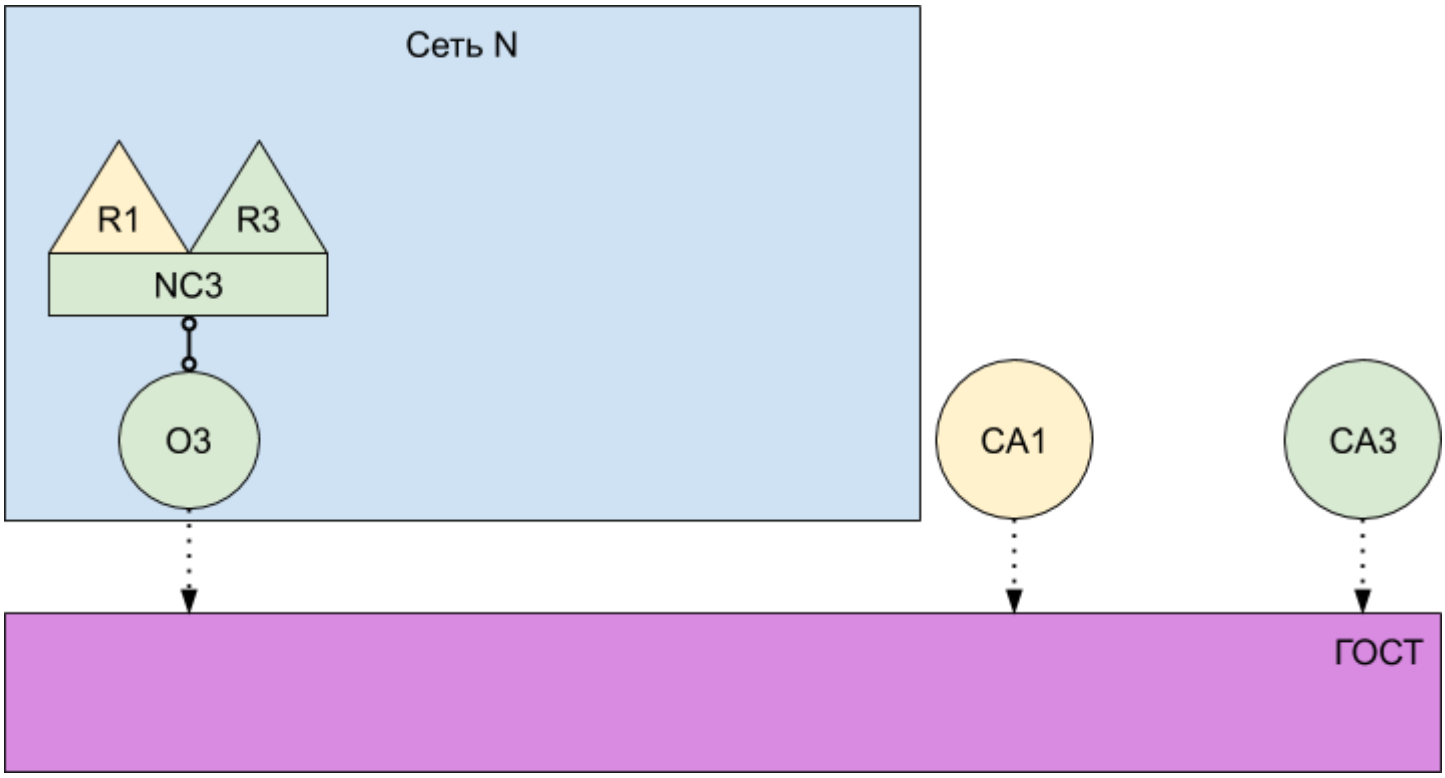
Стоит упомянуть, что сертификаты лежат в основе процесса генерации и проверки транзакции. В частности, они используются в запросах на одобрение транзакции клиентских приложений и ответах смарт-контракта для цифровой подписи транзакции. Сетевые узлы (ноды), на которых размещаются копии реестра (леджера), проверяют действительность подписей по транзакции прежде чем принимать транзакции в леджер.

Итак, опишем еще раз базовую структуру нашего примера сети распределенного реестра. Существует сеть N, доступ к которой осуществляется группой пользователей, определенных центром сертификации CA3. Эти пользователи имеют набор прав в сети N, как описано в политике сети (содержащейся в конфигурации сети NC3). Сеть начинает

работать в тот момент, когда мы запускаем ордерер O3 (узел службы формирования новых блоков).

3.3. Добавление сетевых администраторов

Первоначально конфигурация сети NC3 была настроена на разрешение административных прав только пользователям организации R3. На следующем этапе мы собираемся разрешить пользователям организации R1 администрировать сеть. Давайте посмотрим, как развивается сеть:



Организация R3 обновляет конфигурацию сети NC3, чтобы сделать организацию R1 администратором. После этой точки R3 и R1 имеют равные права на конфигурацию сети.

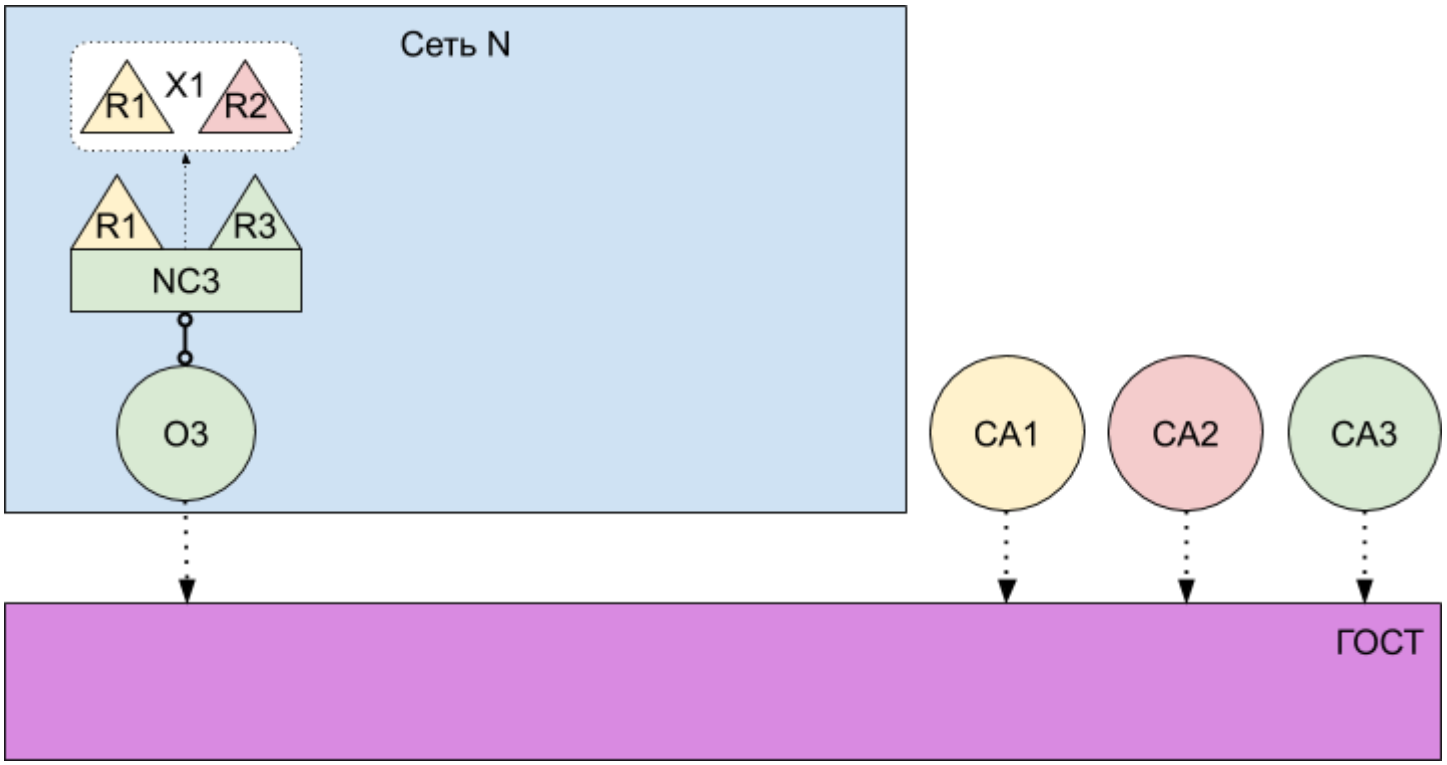
Мы добавили новую организацию R1 в качестве администратора, теперь R3 и R1 имеют равные права в сети. Как видно на схеме, также был добавлен еще один центр сертификации CA1, его можно использовать для идентификации пользователей из организации R1. Центр сертификации CA1, подобно CA2 и O3 использует обращение к серверу для формирования и проверки квалифицированных сертификатов в соответствии со стандартом ГОСТ. После идентификации в сети и подтверждения криптометериалов, пользователи из R3 и R1 могут администрировать сеть.

Несмотря на то, что узел O3 запущен в инфраструктуре R3, организация R1 имеет общие административные права на него до тех пор, пока имеет доступ к сети. Это означает, что и R3, и R1 могут обновить конфигурацию сети NC3, чтобы дать доступ организации R1 к подмножеству сетевых операций.

В своей простейшей форме служба Ordering Service представляет собой один узел в сети, и это показано на примере. Службы Ordering Service обычно являются многоузловыми, они могут быть настроены на разные узлы в разных организациях. Например, можно запустить O3 в организации R3 и подключить его к O1, отдельному узлу организации R1. Таким образом, мы получили бы более сложную многоузловую административную структуру.

3.4. Создание Консорциума

Хотя сеть теперь может администрироваться организациями R3 и R1, в такой сети пока мало что можно сделать. Для начала необходимо определить консорциум. Это слово буквально означает “группа с общей судьбой”, поэтому оно является подходящим выбором для ряда организаций объединяющихся в сеть на Блокчейн платформе “Фабрик”.



Сетевой администратор определяет консорциум X1, в состав которого входят два участника - организации R1 и R2. Это определение консорциума хранится в конфигурации сети NC3 и будет использоваться на следующем этапе развития сети. CA1 и CA2 соответственно являются центрами сертификации для организаций R1 и R2.

Из-за способа настройки конфигурации сети NC3 только R3 или R1 могут создавать новые консорциумы. На этой диаграмме показано добавление нового консорциума X1, который определяет R1 и R2 в качестве учредительных организаций. Для идентификации пользователей из организации R2 был добавлен новый центр сертификации CA2, который также использует проверку криптоматериалов в соответствии со стандартами ГОСТ. Стоит обратить внимание на то, что в консорциуме может быть любое количество

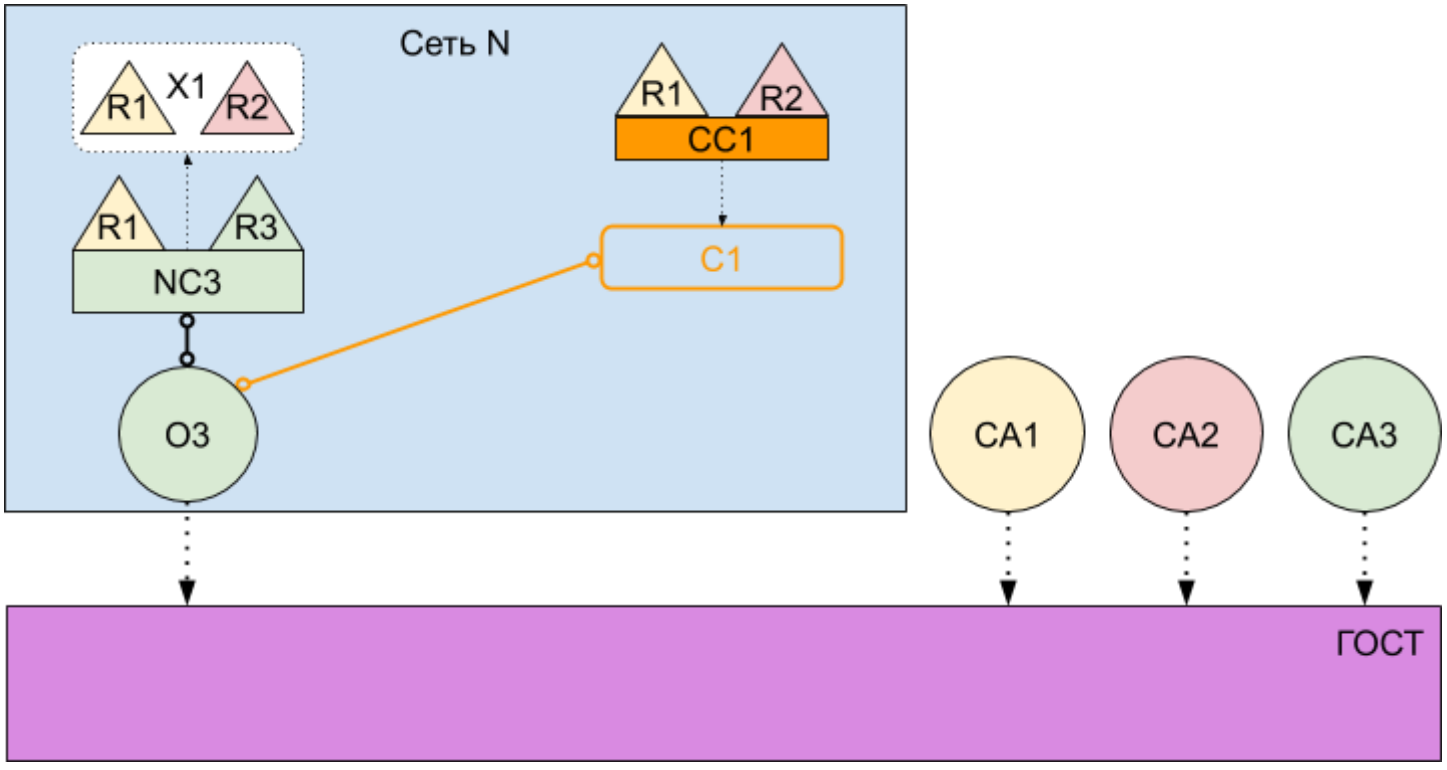
членов организации - в данном примере были использованы только две организации, поскольку это самая простая конфигурация.

Как видно из примера, консорциум определяет набор организаций в сети, которые разделяют необходимость осуществлять операции друг с другом - в данном случае R1 и R2. Имеет смысл объединить организации, если у них есть общая цель, и это именно то, что происходит при создании консорциума.

Данная сеть, хоть и была запущена одной организацией R3, теперь контролируется большим числом организаций. При создании сети можно было сразу назначить организациям R3, R1 и R2 общий доступ, однако, данный пример был намеренно использован для упрощения понимания взаимодействия между организациями в сети.

3.5. Создание канала для Консорциума

Итак, давайте создадим ключевую часть сети - канал. Канал - это основной механизм связи, с помощью которого члены консорциума могут общаться друг с другом. В сети может быть несколько каналов, но сейчас мы начнем с одного.



Канал C1 был создан для организаций R1 и R2 с использованием консорциума X1. Канал управляется конфигурацией канала CC1, полностью отдельной от конфигурации сети NC3. CC1 управляется R1 и R2, которые имеют равные права над каналом C1. R3 вообще не имеет никаких прав в конфигурации CC1.

В процессе создания канала C1 происходит обмен данными, подписями и сертификатами с идентификацией участников системы. Для анализа используемых сертификатов

проводится исследование логов ордерера O3 (узла службы формирования новых блоков) на соответствие стандарту ГОСТ.

Канал C1 обеспечивает частный механизм связи для консорциума X1. Мы видим, что канал C1 был подключен к ордереру O3 (узлу службы формирования новых блоков), но больше к нему ничего не подключено. На следующем этапе развития сети мы собираемся подключить такие компоненты, как клиентские приложения и одноранговые узлы. Но на данный момент канал лишь представляет потенциал для будущего подключения.

Несмотря на то, что канал C1 является частью сети N, он вполне независим от нее. Также обратим внимание, что организация R3 не находится в этом канале - этот канал предназначен для обработки транзакций исключительно между организациями R1 и R2. На предыдущем этапе мы увидели, как R3 может предоставить организации R1 разрешение на создание новых консорциумов. Отметим, что именно организация R3 позволила R1 создавать каналы. К каналу может быть подключено любое количество организаций - в примере были использованы две, так как это самая простая конфигурация.

Канал C1 имеет совершенно отдельную конфигурацию - CC1 - отличную от конфигурации сети NC3. CC1 содержит политики, которые управляют правами взаимодействия между R1 и R2 по каналу C1. При этом организация R3 не имеет доступа к этому каналу. R3 может взаимодействовать с каналом C1, только если одна из организаций канала (R1 или R2) добавит ее к соответствующей политике в конфигурации канала CC1. Примером является определение того, кто может добавить новую организацию в канал: организация R3 не может добавить себя в канал C1 - она должна и может быть авторизована только R1 или R2.

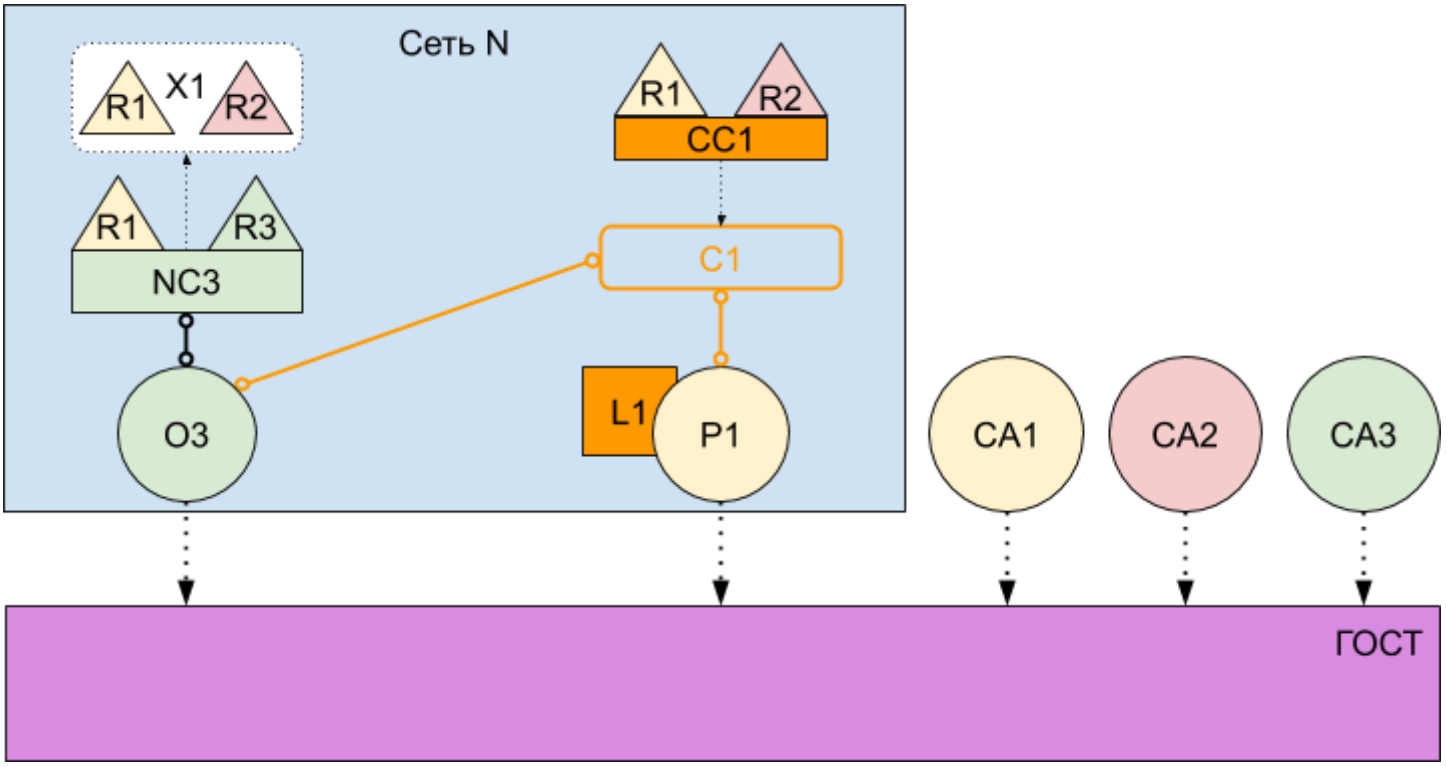
Каналы полезны, потому что они обеспечивают механизм для частных коммуникаций и обменом личных данных между членами консорциума. Каналы обеспечивают конфиденциальность не только от других каналов, но и от сети. Блокчейн платформа “Фабрик” обладает обширными возможностями в этом сегменте, поскольку позволяет организациям совместно использовать инфраструктуру и одновременно сохранять ее конфиденциальность. Здесь нет противоречия - разные консорциумы в сети будут нуждаться в соответствующем обмене различной информацией и процессами, а каналы обеспечивают эффективный механизм для этого. Каналы обеспечивают эффективное совместное использование инфраструктуры, сохраняя конфиденциальность данных и коммуникаций.

Итак, как только канал создан, он в реальном смысле “свободен от сети”. Только организации, которые явно указаны в конфигурации канала, имеют какой-либо контроль над ним. Аналогично, любые обновления конфигурации сети NC3, начиная с этого времени, не будут иметь прямого влияния на конфигурацию канала CC1; например, если консорциум X1 будет изменен, это не повлияет на членов канала C1. Поэтому каналы

полезны, потому что они предоставляют частную связь между организациями, составляющими канал. Кроме того, данные в канале полностью изолированы от остальной части сети, включая другие каналы.

3.6. Узлы(пиры) и Реестры (леджеры)

Давайте теперь начнем использовать канал для соединения сети распределенных реестров и организационных компонентов. На следующем этапе развития сети мы видим, что наша сеть N только что приобрела два новых компонента: пир (узел) P1 и экземпляр распределенного реестра (леджер) L1.



Пир P1 присоединился к каналу C1. P1 физически размещает копию реестра (леджера) L1. P1 и O3 могут связываться друг с другом, используя канал C1.

Одноранговые узлы (пиры) - это сетевые компоненты, в которых хранятся копии распределенного реестра (блокчейна). Цель P1 в сети состоит в том, чтобы просто разместить копию распределенного реестра (леджера) L1 для доступа другим. Мы можем думать о L1 как о физически размещенном на пире P1, но логически размещенном именно на канале C1. Эта идея будет более понятна, когда к каналу подключится большее количество пиров.

Ключевой частью конфигурации P1 является сертификат, выданный центром сертификации, который связывает узел(пир) P1 с организацией R1. Как только узел(пир) P1 запущен, он может присоединиться к каналу C1, используя ордерер O3 (узел службы формирования новых блоков). Когда O3 получает этот запрос на присоединение, он использует конфигурацию канала CC1 для определения разрешений пира P1 на этом

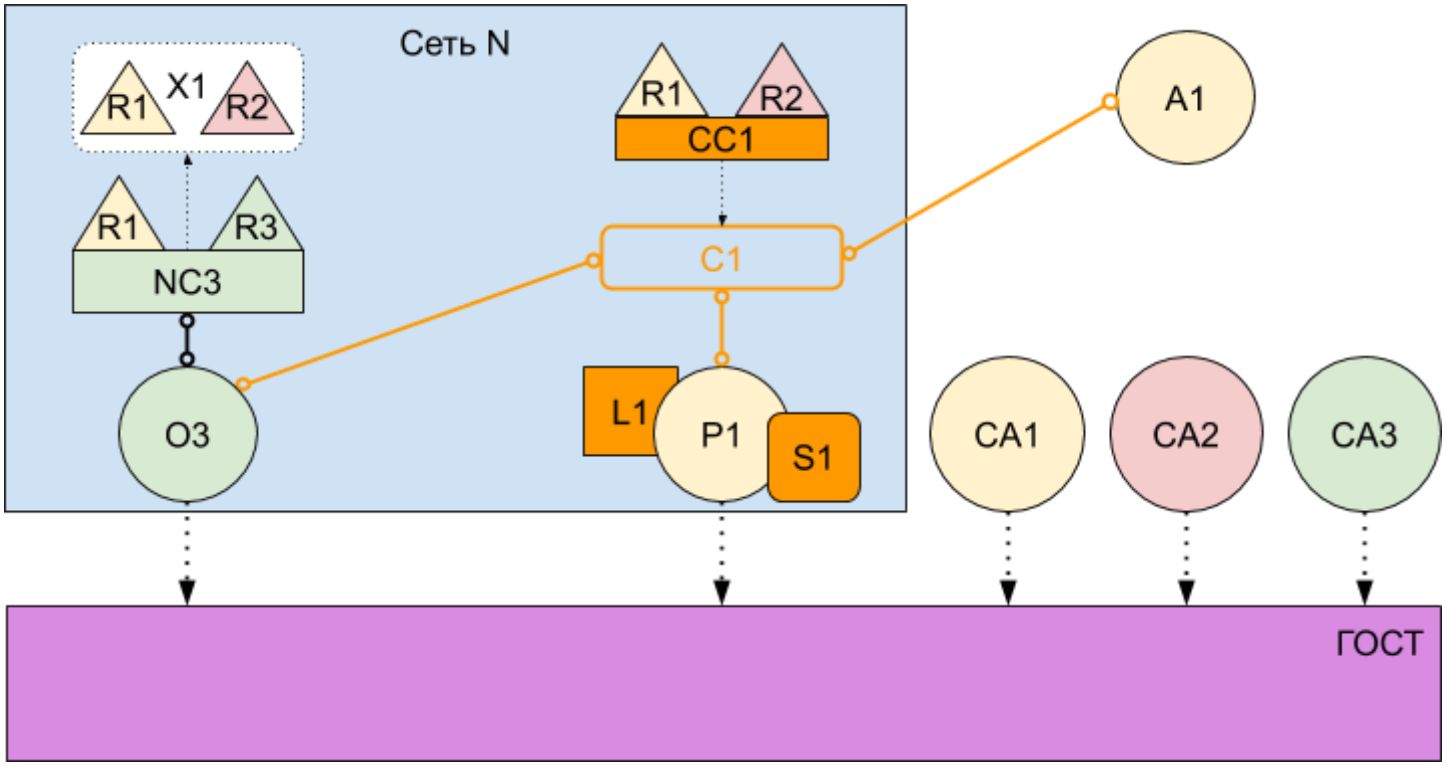
канале. Например, CC1 определяет, может ли пир P1 считывать и / или записывать информацию в реестр L1.

Для анализа используемых сертификатов проводится исследование логов Узла(пира) P1 (подобно ордереру O3) на соответствие стандарту ГОСТ.

3.7. Приложения и чейнкод

Теперь, когда на канале C1 есть леджер, мы можем начать подключать клиентские приложения, чтобы использовать некоторые из сервисов, предоставляемых леджером.

Обратите внимание, как выросла сеть:



Смарт-контракт S1 был установлен на пир P1. Клиентское приложение A1 в организации R1 может использовать S1 для доступа к леджеру L1 через пир P1. A1, P1 и O3 все присоединены к каналу C1, то есть все они могут использовать средства связи, предоставляемые этим каналом.

На следующем этапе развития сети мы видим, что клиентское приложение A1 может использовать канал C1 для подключения к определенным сетевым ресурсам - в этом случае A1 может подключаться как к Узлу(пиру) P1, так и к ордереру O3 (узлу службы формирования новых блоков). Опять можно проследить, как каналы играют центральную роль в коммуникации между сетью и ее компонентами. Подобно взаимодействию между Узлами (пирами) и ордерерами (узлами службы формирования новых блоков), клиентское приложение будет иметь связь с организацией. В нашем примере клиентское

приложение A1 связано с организацией R1; и хотя приложение A1 находится за пределами сети N, оно подключено к организации через канал C1.

Теперь может показаться, что приложение A1 может получить доступ к реестру L1 напрямую через пир P1, но на самом деле весь доступ управляется с помощью специальной программы, называемой чейнкодом смарт-контракта. Смарт-контракт S1 определяет общие шаблоны доступа к леджеру. S1 предоставляет четко определенный набор способов, с помощью которых леджер L1 может быть вызван или обновлен. Проще говоря, клиентское приложение A1 должно пройти через смарт-контракт S1, чтобы иметь доступ к леджеру L1.

Разработчики приложений в каждой организации могут создавать чейнкод для реализации бизнес-процесса, который будет совместно использоваться членами консорциума. Смарт-контракты используются для создания транзакций, которые впоследствии могут быть распределены на каждый узел в сети. На данный момент важно понять, что для корректной работы необходимо выполнить две операции со смарт-контрактом: он должен быть сперва установлен, а затем запущен.

3.7.1. Установка смарт-контракта

После разработки смарт-контракта S1 администратор в организации R1 должен установить его на пир P1. После того, как это произошло, пир P1 может видеть реализацию логики S1 - программный код , который он использует для доступа к леджеру L1.

Когда организация имеет несколько пиров в канале, она может выбрать пир, на который она устанавливает смарт-контракты, нет необходимости устанавливать смарт-контракты на всех пирах.

3.7.2. Запуск смарт-контракта на канале

Однако того, что на пире P1 был установлен смарт-контракт S1, еще недостаточно. Другие компоненты, подключенные к каналу C1, еще не знают о существовании S1; сначала его нужно запустить на канале C1. В нашем примере, который имеет только один пир P1, администратор организации R1 должен запустить экземпляр смарт-контракта S1 на канале C1 с использованием Узла (пира) P1. После запуска экземпляра смарт-контракта каждый компонент на канале C1 знает о существовании S1; для нашего примера это означает, что теперь смарт-контракт S1 может быть вызван клиентским приложением A1.

Обратите внимание, что хотя каждый компонент в канале теперь может получить доступ к S1, он не может видеть его программную логику. Эта информация остается приватной в рамках того узла, на который был установлен смарт-контракт. Установка смарт-контракта

показывает, что он физически размещен на Узле (пире), в свою очередь запуск смарт-контракта показывает, что он логически размещен на канале.

3.7.3. Политика одобрения

Наиболее важной частью дополнительной информации, предоставляемой при запуске смарт-контракта, является политика одобрения. В ней описывается, какие организации должны одобрить транзакции, прежде чем они будут приняты другими организациями в свою копию реестра (леджера). В нашей тестовой сети транзакции могут приниматься в леджер L1, тогда и только тогда, когда транзакция будет одобрена одной из организаций (R1 или R2).

Политика одобрения прописывается в конфигурации канала CC1; это позволяет ей быть доступной для любого участника канала C1.

3.7.4. Вызов смарт-контракта

После того, как смарт-контракт был установлен на Узле (пире) и запущен на канале, он может быть вызван клиентским приложением. Клиентские приложения делают это следующим образом:

1. Происходит отправка предложения по транзакциям (transaction proposals) на Узлы (пиры), которые принадлежат соответствующим организациям, указанным в политике одобрения смарт-контракта

2. Предложение по транзакции служит входом для смарт-контракта, который использует его для генерации одобренного ответа транзакции

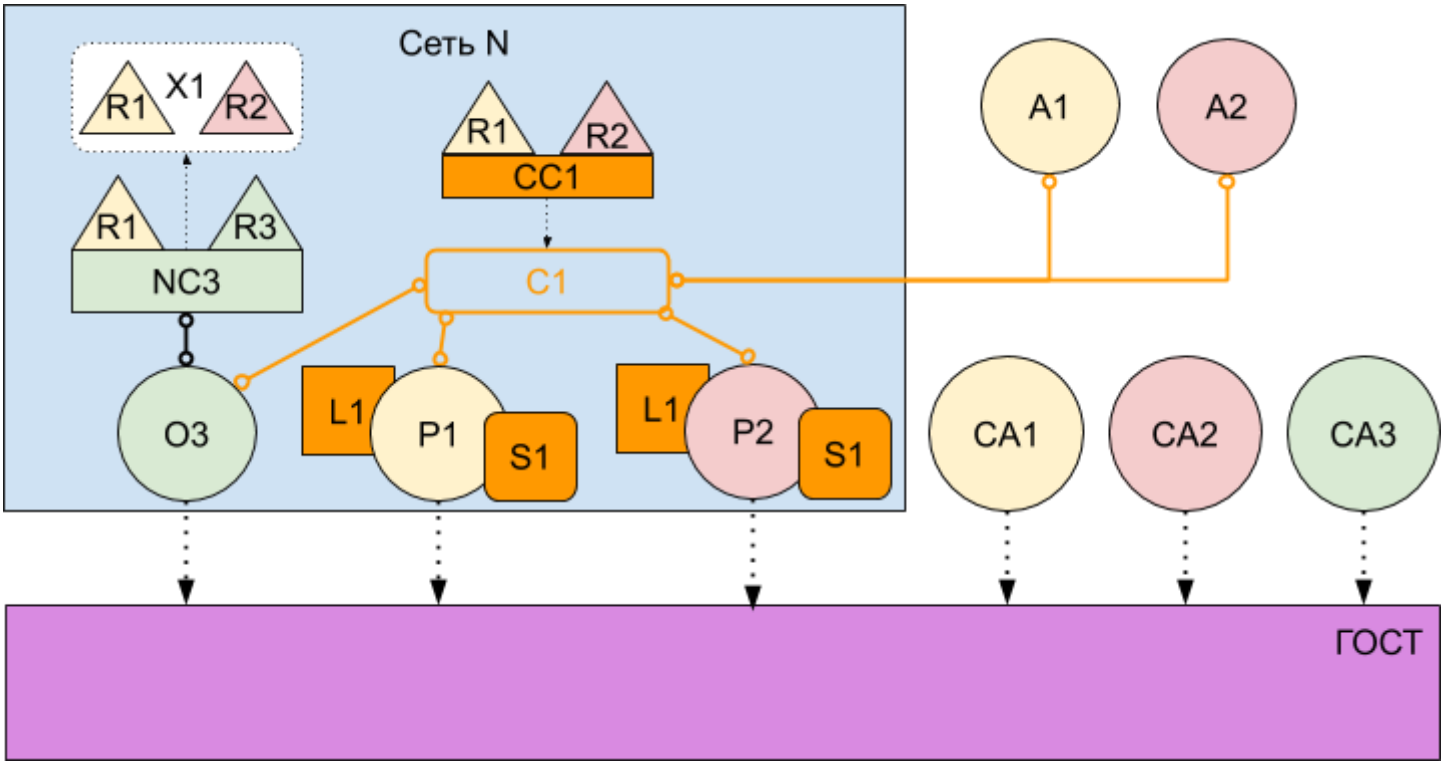
3. Одобренный ответ по транзакции, в свою очередь, возвращается Узлом (пиром) клиентскому приложению.

Именно эти ответы транзакций упаковываются вместе с предложением по транзакции, чтобы сформировать полностью подтвержденную транзакцию. Только подтвержденная транзакция может быть распространена по всей сети.

На этом этапе развития сети мы видим, что организация R1 полностью участвует в сети. Её приложения, начиная с A1, могут обращаться к реестру L1 через смарт-контракт S1, чтобы генерировать транзакции, которые будут подтверждены R1 и, следовательно, приняты в реестр, поскольку они соответствуют политике одобрения.

3.8. Завершение настройки сети

Нашей целью было создание канала C1 для консорциума X1 - организаций R1 и R2. На следующем этапе развития сети организация R1 добавляет свою инфраструктуру в сеть.



Сеть выросла за счет добавления инфраструктуры от организации R2. В частности, в R2 был добавлен новый узел (пир) P2, на котором размещен леджер L1 и чейнкод S1. Узел (Пир) P2 присоединился к каналу C1, как и приложение A2. A2 и P2 идентифицируются с использованием сертификатов центра сертификации C2. Узел (Пир) P2, как и все описанные ранее элементы системы, использует обращение к серверу для проверки сертификатов в соответствии со стандартом ГОСТ. Все это означает, что оба приложения A1 и A2 могут вызывать смарт-контракт S1 на канале C1 используя как при P1, так и узел (пир) P2.

Мы видим, что организация R2 добавила узел (пир) P2 на канал C1. P2 также размещает копию реестра (леджера) L1 и смарт-контракта S1. Мы видим, что организация R2 добавила клиентское приложение A2, которое может подключаться к сети через канал C1. Для этого администратор в организации R2 создал узел (пир) P2 и подключил его к каналу C1, тем же путем, как и до этого администратор в организации R1.

Создание первой сети на Блокчейн платформе "Фабрик" можно считать завершенным. На этом этапе развития сети есть канал C1, по которому организации R1 и R2 могут полностью взаимодействовать друг с другом. В частности, это означает, что приложения A1 и A2 могут генерировать транзакции с использованием смарт-контракта S1 и распределенного реестра L1 на канале C1.

3.9. Резюме сети

В этом разделе было представлено, каким образом различные организации могут интегрироваться в блокчейн-сети на Блокчейн платформе "Фабрик", и как им совместно использовать свою инфраструктуру. Взаимодействие между партнерами может быть заключено в каналы, которые обеспечивают частные механизмы связи между организациями и управляются независимо от общей сети.

Таким образом, Блокчейн платформа "Фабрик" предоставляет возможность комфортно взаимодействовать в одной сети как партнерам, так и конкурентам. Принадлежность к определенной организации любых субъектов, будь то клиентские приложения, Узлы (пиры) или ордереры (узлы службы формирования новых блоков), осуществляется благодаря использованию ими сертификатов из соответствующих центров сертификации. В свою очередь, каждый созданный / выданный сертификат или ключ проходит проверку на соответствие стандартам ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012.

4. Глоссарий

Блок

Блок содержит упорядоченный набор транзакций. Он криптографически связан с предыдущим блоком и, в свою очередь, связан с последующими блоками. Первый блок в такой цепочке блоков называется генезис-блоком или нулевым блоком. Блоки создаются службой Ordering Service и проверяются Узлами (пирами).

Канал

Канал - это приватное перекрытие блокчейна, которое обеспечивает изоляцию и конфиденциальность данных. Распределенный реестр совместно используется равноправными узлами канала, и для взаимодействия с ним транзакционные стороны должны быть надлежащим образом идентифицированы в канале.

Консорциум

Консорциум - это группа организаций блокчейн-сети. Они формируют новые каналы, либо присоединяются к уже существующим. Хотя блокчейн-сеть может иметь несколько консорциумов, большинство сетей используют один консорциум. Во время создания канала все организации, добавленные в канал, должны быть частью консорциума. Однако организация, которая не определена в консорциуме, может быть добавлена к существующему каналу.

Леджер (распределенный реестр)

Распределенный реестр состоит из двух отдельных, хотя и взаимосвязанных компонентов: цепочки блоков и их состояний, также известных как “World state”. Цепочки блоков являются неизменяемыми, обладают иммунитетом к любым

несанкционированным изменениям, то есть после добавления блока в цепочку блок нельзя изменить. В то же время, компонент “World state” содержит текущее значение набора пар ключ-значение, которые были добавлены, изменены или удалены набором валидных и подтвержденных транзакций блокчейн-сети.

Участник сети (Нода)

Нода (узел) - это любой компьютер, подключенный к блокчейн-сети. Участники (Ноды) децентрализованной сети контактируют посредством P2P-протоколов для обмена информацией о блоках и транзакциях. Участник (Нода), в зависимости от типа, хранит только часть или все данные распределенного реестра.

Организация

Организация - это участник сети распределенных реестров. Организация может сама создать сеть или присоединиться к уже существующей, согласно установленным правилам сети. Организация может представлять собой как огромную корпорацию, так и частное лицо. Несколько организаций могут образовать Консорциум. Хотя все организации являются членами сети, не каждая организация будет частью консорциума, что предоставляет возможность находиться в одной сети как партнерам, так и конкурентам.

Служба формирования новых блоков (Ordering Service)

Это распределенная на нескольких узлах служба, которая отвечает за формирование новых блоков и упорядочивание транзакций в блок. Служба Ordering Service является общей связью для всей сети, она содержит криптографические идентификационные материалы, привязанные к каждому участнику.

Узел (Пир)

Сетевой объект, которых размещает на себе копию распределенного реестра (леджера) и запускает чейнкод для выполнения операций чтения/записи относительно реестра (леджера). Каждый узел (пир) должен принадлежать конкретной организации.

Политика одобрения (Endorsement Policy)

Определяет необходимый набор узлов на канале, которые должны выполнять транзакции, присоединенные к конкретному чейнкод-приложению, и проверяет комбинацию ответов (индоссамент). Политика может требовать, чтобы транзакция была подтверждена минимальным количеством подтверждающих узлов, минимальным процентом подтверждающих узлов или всеми подтверждающими узлами конкретного чейнкод-приложения.

Политики могут курироваться на основе приложения и желаемого уровня устойчивости против неправильного поведения (преднамеренного или нет) со стороны одобряющих

коллег. Отправленная транзакция должна соответствовать политике одобрения, прежде чем она будет помечена как действительная путем фиксации одноранговых узлов. Также требуется отдельная политика одобрения для установки и создания транзакций.

Смарт-контракт (чейнкод)

Смарт-контракт - это код, который вызывается клиентским приложением, внешним по отношению к блокчейн-сети. Он управляет доступом и модификацией наборов пар ключ-значение в World State. В Блокчейн платформа “Фабрик” смарт-контракты называются чейнкодом. Чейнкод устанавливается на узлы (пиры) и запускается на одном или нескольких каналах.

Транзакция

Транзакция - это запись новых данных в леджер. Транзакция инициируется пользовательским приложением и заканчивается записью в распределенный реестр.

Центр сертификации (Certification Authority)

Блокчейн платформа "Фабрик" использует такую модель блокчейн-сети, в которой все участники должны быть идентифицированы. Для идентификации используются сертификаты, которые выдают центры сертификации. Каждый созданный / выданный сертификат или ключ проходит проверку согласно внутренним правилам с применением ГОСТ-стандартов.

Цепочка распределенного реестра

Журнал (лог) транзакций, структурированный в виде блоков транзакций, связанных хешем. Узлы (Пиры) получают блоки транзакций от службы Ordering Service, помечают транзакции блока как валидные или невалидные на основе политики одобрения, а также добавляют блок в цепочку хеширования в файловой системе узла (пира).

Служба идентификации (Membership Service)

Это служба, которая проверяет подлинность, авторизует и управляет идентификацией. Она относится к абстрактному компоненту системы, который предоставляет учетные данные клиентам и их партнерам для участия в сети Блокчейн платформы “Фабрик”. Именно через нее участники сети осуществляют проверку принадлежности объекта к той или иной организации или каналу.

World State

Компонент распределенного реестра, который хранит текущие (последние) значения всех объектов распределенного реестра.

5. Системные требования для установки демонстрационной сети

5.1. Аппаратное обеспечение

Окружение	Требования
Процессор	Intel Xeon Processor (Skylake, IBRS), 2 Cores
Память	4 Гб
Объем диска	20 Гб

5.2. Операционные системы

Ubuntu 18.04+

5.3. Безопасность

Linux/UNIX: Для работы тестовой сети необходима установка под учетной записью супер-пользователя.

6. Подготовка системы к установке

6.1. Установка пакетов Docker и Docker-compose

Каждый компьютер сети должен иметь доступ к глобальной сети Интернет. На компьютере сети должны быть установлены:

- docker 18.06.1-ce+
- docker-compose 1.22.0+

Для установки docker на необходимо использовать команды:

```
$ sudo apt-get update
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
```

```
    curl \
    gnupg-agent \
    software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Детали установки находятся на официальном сайте Docker:
https://docs.docker.com/install/linux/docker-ce/ubuntu/

Для установки *docker-compose* нужно использовать команды:

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(una
me -s)-$(uname -m)" -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose
```

Детали установки находятся на официальном сайте Docker:
https://docs.docker.com/compose/install/

Пользователь, запускающий скрипты, должен быть участником docker группы. Команда для проверки:

```
user@host:~$ groups

user docker
```

Если пользователь не в группе docker, нужно выполнить команду

```
$ sudo usermod -aG docker USER_NAME
```

где USER_NAME название учетной записи пользователя системы линукс, от которого будет запускаться приложение.

6.2. Распаковка докер-образов

Для запуска скриптов необходимо скачать архив "install.zip", размещенный по адресу http://фабрик.рф/install.zip. Распакуйте архив командой

```
unzip ./install.zip
```

Внутри архива содержатся docker-образы:

1. bg11_server.tar.gz – сервис криптографии;
2. fabric-ca-gost.tar.gz – сервис управления ключами;
3. fabric-ccenv-gost.tar.gz – сервис демонстрационного приложения;
4. fabric-orderer-gost.tar.gz – сервис службы Ордерер;
5. fabric-peer-gost.tar.gz – сервис узла;
6. fabric-tools-gost.tar.gz – инструменты.

Выполните команды для установки требуемых docker-образов:

```
docker load --input ./bg11_server.tar.gz
docker load --input ./fabric-ca-gost.tar.gz
docker load --input ./fabric-ccenv-gost.tar.gz
docker load --input ./fabric-orderer-gost.tar.gz
docker load --input ./fabric-peer-gost.tar.gz
docker load --input ./fabric-tools-gost.tar.gz
```

6.3. Конфигурация сети

Конфигурационные файлы не требуют изменений, для запуска тестовой сети можно запускать их с настройками по умолчанию.

В особых случаях некоторые настройки по умолчанию могут быть изменены. Конфигурации находятся в файлах *config/{config_group}.sh*. Эти файлы содержат значения по умолчанию. Для редактирования какого-либо значения скопируйте соответствующий файл *config/user.{config_group}.sh.template* в *config/user.{config_group}.sh* и отредактируйте его содержимое.

Основные переменные конфигурации находятся в *config/user.common.sh.template* с комментариями.

6.4. Запуск сети

Скачайте приложение *gost-hello.tar.gz* в виде архива в любую папку. Распакуйте архив в той же папке с помощью команды:

```
tar -xf gost-hello.tar.gz
```

Выполните в этой папке команды:

1. Настройка среды

```
./setup.sh
```

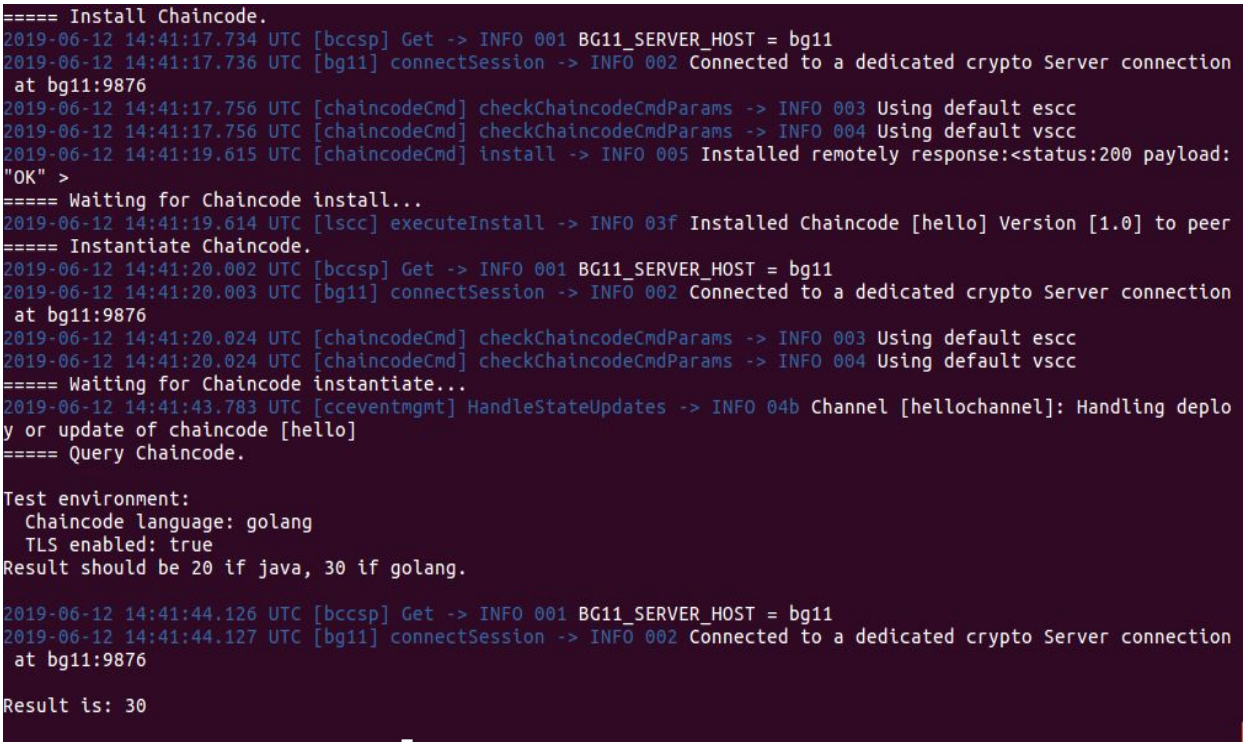
2. Запуск докер контейнеров

```
./start.sh
```

3. Запуск дефолтных тестов

```
./run-test.sh
```

Успешно выполненные тесты в консоли выглядят следующим образом:



6.5. Остановка сети

Все узлы сети могут быть отключены с помощью следующей команды:

```
./stop.sh
```

Процесс остановки сети в консоли выглядит следующим образом:

```
Stopping tools ... done

Removing tools ... done

Network hellonet is external, skipping
Stopping org0-peer0 ... done

Stopping org0-peer0-couchdb ... done

Removing org0-peer0 ... done

Removing org0-peer0-couchdb ... done

Network hellonet is external, skipping
dev-org0-peer0-hello-1.0
Untagged: dev-org0-peer0-hello-1.0-680d0876e12b5b9cbacb1d4218781caa94c934fcc921eae325a453c49a50c526:latest
Deleted: sha256:fd9ebb9311e805b97157efea90521db9b9045ff445b275dda92dd7ef7f9eca05
Deleted: sha256:5338ff0d3d3a6cba7dc691994ab1db1aeaf8aaf7eb0f58da7fca986ce34f210c
Deleted: sha256:5a5d8feb05b074419bf4f75f9e9e0b8647cd0af2d4c9a184a74a2f909e4e5fe0
Deleted: sha256:11f3e8ad0be316e4451f965a1a60a4a813bbb1fd068128f9d109fa7e49bfa442
Untagged: dev-org0-peer0-hello-1.0-c55cd3c03af462cbe69aecb2b3528c60c459995f46a775ae6520b3f9a2a3342c:latest
Deleted: sha256:8f4f4efb887a21cd7072e04f76f5442fe4d4c6a070811acd4f9ad666089ddcc2
Deleted: sha256:beb6a9a8d5b4a7e7b0313f0aee60adf803081d1d4facda17d94fa84f9672c853
Deleted: sha256:51302daf1bbe248c8f202ed62cb8aa83c4b7e66838a112520447f200246c70a3
Deleted: sha256:edf2d77c5c8e131225e0e680b0c01ea9479fed7411d40b30205d21586f31065c
Stopping orderer ... done

Removing orderer ... done

Network hellonet is external, skipping

===== Stopping MAIN CA =====
WARNING: The BG11_SERVER_HOST variable is not set. Defaulting to a blank string.
WARNING: The USE_GOST variable is not set. Defaulting to a blank string.
WARNING: The USE_TLS variable is not set. Defaulting to a blank string.
WARNING: The HOST_USER_ID variable is not set. Defaulting to a blank string.
WARNING: The HOST_GROUP_ID variable is not set. Defaulting to a blank string.
WARNING: The DOCKER_IMAGE_CA variable is not set. Defaulting to a blank string.
Stopping main-ca ... done

Removing main-ca ... done

Network hellonet is external, skipping

===== Stopping BG11 =====
Stopping bg11 ... done

Removing bg11 ... done

Network hellonet is external, skipping
```

7. Устранение неполадок

В случае возникновения неполадок при установке тестовой сети выполните следующие действия:

1. Зайти в папку или создать её, где будут собраны логи docker-a. Местоположение не важно.

2. Создать в этой папке файл `save_gost_hello_logs.sh`.

3. Отредактировать файл и поместить в него содержимое из листинга **7.1** (см. ниже)

4. Сохранить файл. Сделать файл исполняемым с помощью команды:

   ```
   $ chmod +x save_gost_hello_logs.sh
   ```

5. Запустить файл на выполнение командой:

   ```
   $ ./save_gost_hello_logs.sh
   ```

6. После выполнения команды в данной папке будет файл с архивами логов с именем `gost-hello.tar.gz`

7. Отправить данный файл и описание проблемы в службу поддержки

7.1. Листинг файла `save_gost_hello_logs.sh`

```
#!/bin/bash

IMAGES="bg11 main-ca orderer org0-peer0 org0-peer0-couchdb"
FILE_NAMES=""

for IMAGE in $IMAGES
do
  docker logs $IMAGE >& ./$IMAGE.log
  FILE_NAMES="$FILE_NAMES $IMAGE.log"
done

tar -czf gost-hello.tar.gz $FILE_NAMES
```