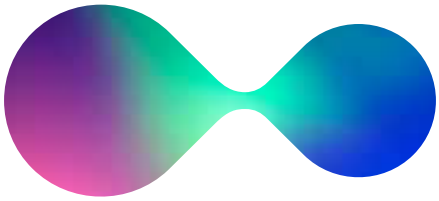


mira

Техническое задание
на разработку платформы

Содержание

1. Общие сведения о продукте

Mira – это децентрализованый сервис, пользовательская часть которого представлена в виде десктопного/мобильного/web-приложения, которое позволяет производить операции с MiraBox(-ами).

MiraLab.io – онлайн сервис для работы с MiraBox(-ами).

MiraWallet – десктопное приложение (macOS, Windows, Linux) для работы с MiraBox(-ами).

MiraWallet Mobile – мобильное приложения для гаджетов на базе iOS и Android.

MiraNet – блокчейн Mira, основанный на Ethereum.

Smart contract – сущность Mira Blockchain, фиксирующая в себе метаинформацию о боксе, и содержащая статус его вскрытия на данный момент.

MRC – валюта сети MiraNet, технически представлена эфиром, используется для оплаты газа в MiraNet, может быть обменена на токен Mira по курсу 1:1.

MIRA token – валюта в MainNet, размещается на биржах и выплачивается инвесторам в процессе проведения ICO. Может быть обменена на MRC по курсу 1:1.

Лаборатория SmartBox(-ов) – отдельный раздел платформы Mira, в котором сторонние разработчики смогут создавать кастомизированные смарт-контракты и предлагать их системе.

MiraBox – это контейнер, состоящий из файла и связанного контракта в MiraNet.

MultiBox – это Тип MiraBox(-а), тело которого состоит из:

- Currency #1 (Одна из валют в контейнере)
- Currency #2 (Одна из валют в контейнере)
- Currency #N (Одна из валют в контейнере)
- File (0<25 MB) - файл любого типа, размером до 25MB

NominalBox – это Тип MiraBox(-а), тело которого состоит из:

- Currency - единственная валюта в контейнере

SmartBox – это MultiBox, связанный с смарт-контрактом. Смарт-контракт, получая информацию от ораклов, может вводить ограничение на открытие содержимого SmartBox до выполнения или наступления определенных внешних условий.

Private MiraBox – имеет минимум открытой информации: идентификационный номер, статус вскрытия и публичные ключи нод, использованных для шифровки.

MiraOracles – это offchain системы, анализирующие данные, которые могут быть использованы как условия открытия SmartBox`а.

MiraNode – это один из компонентов серверной части, который обеспечивает генерацию и хранение ключей от контрактов\кошельков мультиподписи для всех поддерживаемых системой криптовалют.

2. Требования к ПО

2.1. Функциональные требования

Mira представляет открытый многокомпонентный сервис, участники которого могут быть представлены как серверными сущностями, так и пользователями клиентского ПО.

Программное обеспечение серверов должно включать компоненты: MiraNet, MiraNode, MiraOracle, которые могут быть установлены как частично, так и полностью. Владельцы серверов, где будут размещаться указанные выше сервисы, смогут получать доход от выполнения операций, связанных с поддержанием инфраструктуры Mira.

- Владельцы хостов MiraNet (Ethereum Private Network Nodes) - оплата за майнинг транзакций;
- Владельцы серверов с MiraNode - вознаграждение при расшифровке MiraBox;
- Администраторы MiraOracle - выплаты за успешные прогнозы ораклов.

Пользователи могут пользоваться сервисами при помощью браузера (MiraLab), мобильного телефона (MiraWallet Mobile), десктопного компьютера Windows\macOS\Linux (MiraWallet).

MiraBox состоит из файловой и контрактной части. В файле контейнера хранится информация:

- ID MiraBox'а - идентификатор, представленный уникальным ECDSA публичным ключем ;
- Contract –адрес контракта в MiraNet, содержащего информацию и состояние контейнера.
- Public Data – открытая информация соответствующая данному контейнеру.
- Private Data – содержит закодированный симметричным AES-256 ключом ECDSA приватный ключ, парой от которого является ID MiraBox'а.

При создании MiraBox'а происходит деплой соответствующего смарт-контракта, адрес которого записывается в открытое поле "Contract" внутрь файла, и по нему всегда можно проверить статус вскрытия контейнера, а также узнать, какая информация находится внутри.

В контракте хранится:

- Type - тип содержимого:
  - MultiBox - если MiraBox также содержит зашифрованный контент, не являющийся криптовалютами;
  - NominalBox.
- Conditions - условия вскрытия:
  - Box (MiraBox без условий вскрытия);
  - SmartBox (MiraBox с определенными условиями вскрытия).

- Creation Date - дата создания;
- Opening date - дата вскрытия;
- State - статус;
- Nodes - набор адресов нод, участвующих в шифровании;
- Currencies - набор адрес(а) мультиподписи одной или нескольких криптовалют, находящихся в контейнере;
- Hash - sha256-хеш от зашифрованной части файла MiraBox'а, чтобы защитить его от подделки.

MiraBox'ы, содержащие только криптовалюту, имеют номинальную стоимость, соответствующую значению содержимого и называются NominalBox.

При создании MiraBox'а, платформа генерирует публичный ключ от выделенного кошелька, на который можно перевести средства с личного кошелька либо закупить крипто-валюту через платформу за фиатные деньги.

2.2. Требования к серверной части

Рекомендуемые требования к серверному оборудованию:

- 2 GB RAM
- 2 CPU Cores
- 50GB SSD/HDD Storage

Мастер-нода должна включать сервер узла сети Ethereum (например Geth) с корректно настроенной конфигурацией для работы в режиме private network.

Приватная сеть (MiraNet) должна включать все функции Ethereum, в т.ч. поддерживать создание\выполнение смарт-контрактов и взаимодействие с ними.

Второй важный компонент серверной части - сервис MiraNode. Им представлена основная часть функционала серверной части. Сервис обеспечивает генерацию и хранение ключей от контрактов\кошельков мультиподписи для всех поддерживаемых системой криптовалют.

Еще одна серверная сущность, функционирующая как smart-oracle – MiraOracle, представлено сервисом, прослушивающим event-логи сети MiraNet, реагирующим на них проверкой offchain-событий и сообщением их результатов назад в контракт через транзакцию.

Взаимодействие серверных компонентов между друг-другом и их связь с клиентским ПО большей частью осуществляется через смарт-контракт в MiraNet.

3. Описание компонентов системы

3.1. Платежный процессинг

Централизованная Web-система, позволяющая принимать фиатные платежи и пополнять криптовалютой MiraBox'ы. С одной стороны представлена платежным шлюзом, например, таким как www.interkassa.com, с другой блокчейн-minter'ом, начисляющим пользователям MRC в случае успешного проведения платежа.

3.2. MiraWallet (macOS, Windows, Linux):

Предоставляет для пользователя функционал по созданию и распаковке MiraBox'ов. При запуске приложение обновляет реестр мастер нод, и запрашивает статусы ранее созданных контейнеров.

Функционал

1. Создание MiraBox'ов:

Дает пользователю возможность выбрать подходящие настройки, через конфигуратор контейнеров, позволяет создавать как обычные, так и SmartBox'ы.

После ввода характеристик пользователь либо оплачивает через платежный процессинг, либо пополняет криптовалютой самостоятельно, затем бокс регистрируется в блокчейне и пользователь задает пароль, которым шифруется содержимое и приложение выдает MiraBox в виде файла

2. Распаковка:

Чтобы увидеть содержимое MiraBox'а в расшифрованном виде пользователь может просто перенести файл в окно приложения и ввести пароль, а десктопное приложение в свою очередь выполнит запрос в блокчейн Mira.

3.3. Мобильное приложение (iOS, Android)

Дублирует функционал десктопного приложения. Имеет возможности экспорта файлов MiraBox через e-mail и мессенджеры.

3.4. Веб-приложение

Функционал приложения, включая генерацию MiraBox'а должны представлять в большей степени скрипты, выполняющиеся на клиентской части, за исключением обращений к мастер-нодам и блокчейну. Обращение к нодам выполняется через JSON-RPC. Интеграцию с блокчейном MiraNet можно реализовать через Metamask\Mist, настроенный на Mira, либо внешний web3-провайдер. Интерфейс веб-приложения должен быть визуально схож с десктопным, однако верстка должна предполагать адаптивность разметки для поддержки устройств с разными разрешениями дисплея.

По функционалу, также, как и MiraWallet делится на следующие разделы:

1. Конструирование MiraBox'ов, выбор набора криптовалют, для построения номинальных контейнеров.

- Конфигуратор SmartBox'ов, генерация инструкций для ораклов.

2. Просмотр MiraBox'ов.

- Просмотр статуса\метаинформации об открываемом контейнере из файла и блокчейне
- Расшифровка и вывод средств из MiraBox'ов.

3.5. Ораклы + Ноды

Представляют набор микросервисов: Ethereum Private Network Node, MiraOracle, MiraNode.

3.5.1. Ethereum Private Network Node

Обеспечивают все необходимое для полноценной работы частной сети на основе Ethereum включая механизм смарт-контрактов.

3.5.2. MiraOracle

Система, позволяющая децентрализованно выполнять offchain операции, по аналогии с ChainLink и Oraclize. Ораклы в системе являются необходимым источником внешней информации для выполнения SmartBox'ов.

3.5.3. MiraNode

Один из основных компонентов системы. Наравне с пользовательскими приложениями Mira, блокчейн Mira обеспечивает распределенное шифрование/дешифрование контейнеров различных типов.

4. Классы, роли и характеристики пользователей

Роль	Фичи			
	Просмотр и проверка MiraBox'ов	NominalBox	MultiBox	SmartBox
Пользователь	+			
Авторизованный пользователь	+	+	+	+
Токенхолдер. Администратор	+	+	+	+
Токенхолдер. Арбитр	+	+	+	+

Пользователь

Установив приложение, либо перейдя на сайт Miralab.io, любой посетитель без регистрации и входа уже может пользоваться таким функционалом, как:

- Просмотр публичной информации о MiraBox'ах в MiraNet, в т.ч. номинал;
- Проверка целостности и подлинности файлов MiraBox.

Авторизованный пользователь

Регистрация и вход в систему выполняется с помощью ECDSA пары ключей, которые можно импортировать\экспортировать между разными устройствами.

Процесс регистрации представляет собой локальную генерацию пары ключей, которая, в целях безопасности, не передается 3-й стороне, либо микросервисам Mira. Однако, для удобства, пользователь в любой момент может перенести свои ключи на другое устройство с помощью QR-кодов, либо копирования. Авторизованный пользователь имеет доступ ко всем сервисам платформы и может:

- Создавать MiraBox (Nominal\Multi);
- Пользоваться конструктором SmartBox.

В мобильном приложении аккаунт дополнительно может быть защищен коротким цифровым пин-кодом. В iOS приложение может быть защищено Touch ID, Face ID (iPhone X).

Токенхолдер-Администратор

Имеет все возможности авторизованного пользователя, а также возможность организовывать собственную ноду системы Mira с возможностью монетизации ее при помощи:

- предсказаний offchain событий (Oracle);
- майнинга транзакций;
- хранения ключей от MiraBox'ов.

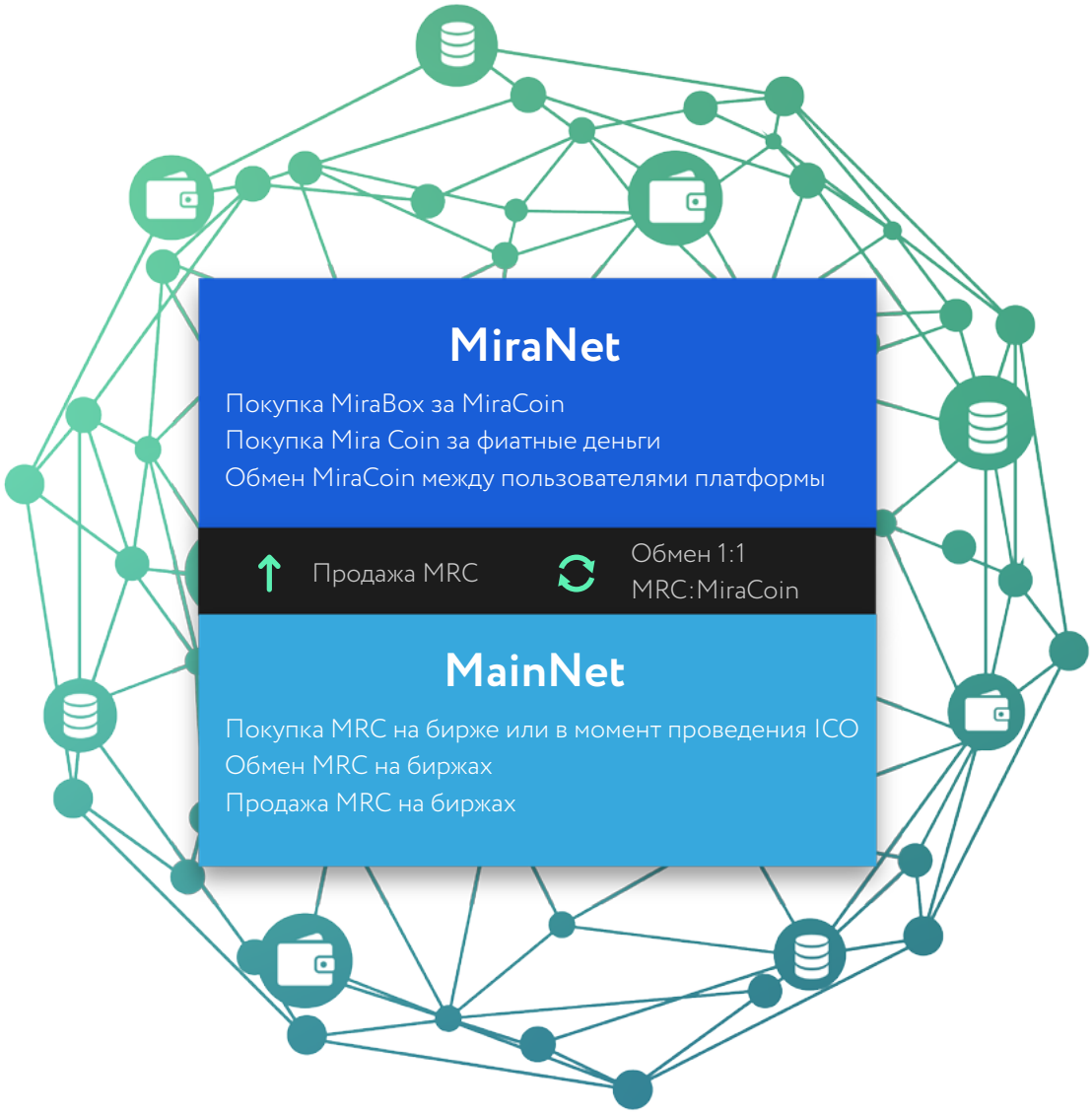
Токенхолдер-Арбитр - сертифицированный платформой Mira пользователь. Имеет все возможности авторизованного пользователя, а также доступ к сделкам, в которых было активировано его участие. Арбитр имеет полномочия для разрешения спорных ситуаций.

5. Внутренний токен платформы Mira

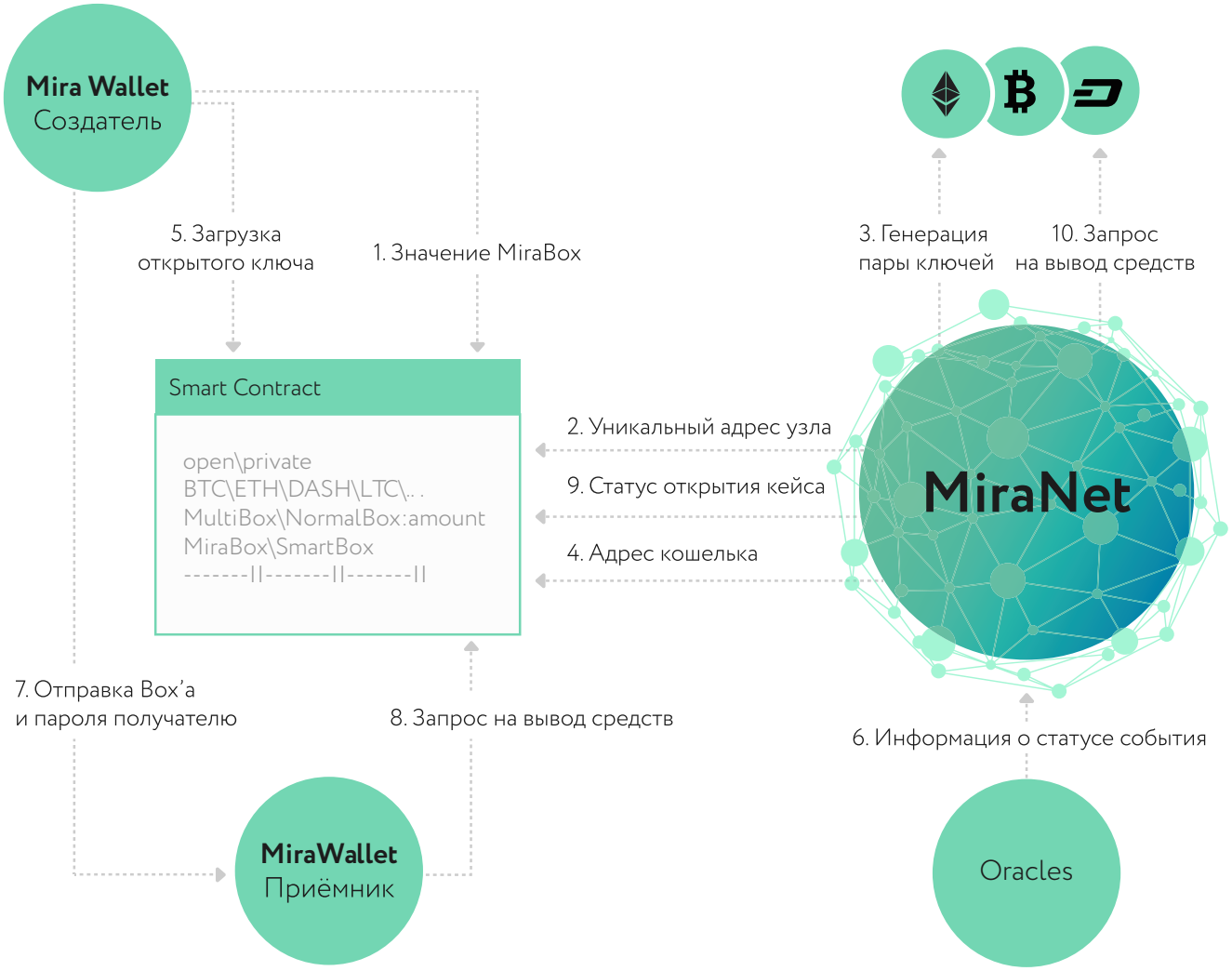
MRC - внутренняя валюта сервиса, структурно представляющая аналог эфира в сети MiraNet. MRC выполняет расчетную функцию, позволяет покупать SmartBox, и, одновременно, выступает в роли оплаты Газа для выполнения смарт-контрактов. MRC может в любой момент быть обменен на MIRA token по курсу 1:1

MRC может быть передан от одного пользователя другому. Предусмотрено два вариант получения MRC(-а):

1. Купить MIRA token на ICO или бирже, затем обменять на MRC;
2. Купить MRC за фиат через платежный шлюз.



6. Схема взаимодействия пользователя с системой



1. Пользователь инициирует создание (Smart) MiraBox в приложении (Десктоп\Веб\Мобильном) и настраивает события, по наступлению которых Box может быть открыт. При этом происходит деплой блокчейн смарт-контракта, отвечающего за конкретный box, во время чего с пользователя списывается оплата в MRC.

Контракт на этом этапе описывает только мета-информацию:

- открытость(open\private);
- содержимое(btc\eth\dash\ltc\...);
- номинальность(MultiBox\NominalBox:amount);

- условия вскрытия(Box\SmartBox);
- и прочие данные - момент создания, идентификатор и т.д.

2. Ноды получают оповещение о новосозданном боксе, адреса нескольких, случайно выбранных нод, сохраняются в смарт-контракте, что является информацией о том, какие из нод могут быть использованы для участия в открытие бокса, за что получат бонус.

3. Вышеупомянутые ноды генерируют и сохраняют пары ключей мультиподписи соответствующей валюты (валют).

4. Адрес кошелька мультиподписи возвращается назад в контракт.

5. Приложение Mira генерирует пару ключей соответствующей валюты, шифрует приватный ключ паролем, загружает публичный ключ, хеш от приватного передается в смарт-контракт.

Если помимо криптовалют в MiraBox'е содержится произвольная информация, т.е. контейнер представляет MultiBox, она также помещается в контракт, в зашифрованном виде. Данные ассиметрично зашифровывается с использованием ECC, публичными ключами, уже сохраненных в контракт на шаге (3) и симметрично с помощью AES-256, пользовательским паролем.

6. Ораклы при наступлении события, которое было задано, если это SmartBox, отмечают об этом в контракте.

7. Создатель бокса передает его, вместе с паролем, получателю.

8. Получатель открывает файл Box`а приложением Mira, вводит полученный вместе с ним пароль, и, если он верный, системе удается извлечь приватный ключ, с помощью которого делается вызов на смарт-контракт, означающий запрос на вывод средств из MiraBox`а.

9. Ноды получают сведения об успешном вскрытии из блокчейна.

10. Ноды, с использованием приватных ключей созданных на шаге (3), выполняют запрос на кошелек мультиподписи для выполнения операции вывода средств. Если все подписи верны, происходит успешный вывод средств. Если это MultiBox с произвольной информацией, то ноды также сообщают в контракт сгенерированные приватные ключи, для ее расшифровки.

7. Безопасность

Платформа Mira представляет децентрализованный сервис, благодаря чему, потенциальный несанкционированный доступ на любые элементы инфраструктуры не даст злоумышленнику доступа к содержимому MiraBox'ов. В качестве криптографических алгоритмов, используемых для шифрования используются: AES-256 и ECC-Secp256k1, что надежно защищает файлы MiraBox от локального брутфорса.

Все компоненты системы полностью представлены opensource - решениями, исходный код которых будет доступен на GitHub и может быть проаудирован любым желающим.

Между микросервисами, API которых представлен в виде REST, например JSON-RPC, взаимодействие осуществляется только по HTTPS с доверенным, подписанным SSL - сертификатом.

7.1. Механизм мультиподписи

Для открытия MiraBox'а с криптовалютой необходимо пройти multisig-верификацию. Это значит, что для выполнения транзакции необходимо более одной ECDSA - подписи.

Таким образом, ни один участник системы не может получить доступ к средствам в односторонем порядке. В тоже время, после вскрытия, MiraBox помечается как использованный, и не может быть вторично использован в MiraWallet

7.2. Двойное шифрование файла MiraBox

При создании MiraBox(-a) через MiraWallet, его содержание зашифровывается с помощью симметричного алгоритма AES-256. Если пользователь создает MultiBox, то дополнительно происходит шифровка с помощью асимметричного ECC (Elliptic-curve cryptography).

7.3. Ораклы

Распределенная система предсказания offchain - событий, построенная на том же принципе, что и сервисы Oraclize и ChainLink. Т.е. события проверяются множеством независимых серверов, имеющих рейтинговую систему репутации, на основе которой происходит PoS (Proof-of-stake) подтверждение предсказания определенного оракла.