Протокол взаимодействия узлов (Жидкая экономика)

Thursday, August 29, 2019 12:16 PM

Все сообщения отправляются последовательно по цепочке взаимодействий от производителей к потребителям и необходимы лишь для синхронизации состояния индивидов.

Обмен QR кодами:

(Поля разделяются пробелами)

Digest
Формируется Потребителем и передается Производителю перед началом взаимодействия.

Полезная нагрузка:

Размер поля	Описание	Тип данных	Комментарии
32 байт	PubKey	строка	Публичный ключ Потребителя
?	Sig from PubKey	Строка	Подпись закрытым ключем дайджеста открытого ключа

Производитель проверяет подлинность подписи и наличие потребителя в своем реестре, если подпись верна и потребитель есть в реестре то производитель отправляет потребителю координаты сигнального сервера через который будет производится синхронизация, токен для идентификации сессии и свой публичный ключ для внесения его потребителем в свою базу или обновления состояния.

CheckKey
Формируется Производителем перед началом оказания услуг.

Полезная нагрузка:

Размер поля	Описание	Тип данных	Комментарии
32 байт	PubKey	строка	Публичный ключ Производителя
40 символов	Socket	string	URL сигнального сервера
?	token	string	Токен для работы с сигнальным сервером

После соединения клиенты отправляют сообщения серверу содержащее токен прозводителя услуг.

Сигнальный сервер должен проанализировать сообщения от подключающихся клиентов, если два клиента сообщили в сообщении секретный токен то сервер создает канал связи между клиентами, в ином случае разрывает связь.

Далее у обоих клиентов запускается таймер который отслеживает последнюю активность в канале связи. Если никаких сообщений не поступает определенное время то приложение отключается от канала. Кроме того после получения QR кода производителя и установки связи с сигнальным сервером, потребитель (его приложение) начинает переодически отправлять запрос в этот канал с типом сообщения getHashs, где pos = 0, что означает что потребитель запрашивает хеш сумму корня базы производителя.

И производитель и потребитель в любой момент могут прервать синхронизацию либо вообще ее не начинать.

Обмен сообщениями:

Примитивы:

MsgType(тип сообщения)

Value	Name	Описание
0	getHashs	Запрос на получение дочерних хешей в древе аккаунтов(отправляет потребитель услуг)

1	hashs	Список дочерних хешей в древе аккаунтов(ответ на запрос)

Сообщения:

Message for node

Field Size	Description	Data type	Comments
1	msgType	uint8_t	Тип сообщения
4	sigLength	uint32_t	Размер подписи
~80?	sig	uchar[]	Signature <= (**MsgType+**pubKey+first4 byte(nodeHash))
?	payload	uchar[]	The actual data

Payloads:
Размер payload байт (maximum 256 entries, which is just over 1.7 megabytes)

GetHashs(массив)
Формируется потребителем, массив позиций дерева (maximum 256 entries, which is just over 2 Kbytes)

Полезная нагрузка:

Размер поля	Описание	Тип данных	Комментарии
8	pos	[uint64_t]	Позиция узла в дереве в котором запрашиваются дочерние хеши

Hashs(массив)
Формируется производителем в ответ на getHashs (maximum 256 entries, which is just over 1.7 megabytes)

Полезная нагрузка:

Размер поля	Описание	Тип данных	Комментарии
8	pos	long	Позиция узла
1	type	byte	Тип узла
1	keySize	byte	Размер ключа
0-18	key	uchar[]	Ключ узла
32	childsMap	uchar[]	Карта дочерних узлов
2-7168	childsArray	[**value**]	Массив данных в запрашиваемом узле

value

Полезная нагрузка(дочерние узлы\ключи):

Размер поля	Описание	Тип данных	Комментарии
2\8	age\pos	uchar[]	Если тип родителя Leaf то возраст ключа, иначе позиция дочернего узла
0-20	hash	char[0-20]	Если тип родителя Branch то возвращается хеш дочернего узла