

Санкт-Петербургский государственный университет
Кафедра компьютерного моделирования и многопроцессорных систем

Максудян Людмила Ареговна

Выпускная квалификационная работа бакалавра

**Реализация системы товарообмена на базе
технологий распределенных реестров**

Направление 02.03.02

«Фундаментальная информатика и информационные технологии»

ООП «Программирование и информационные технологии»

Научный руководитель,

PhD,

доцент

Корхов В. В.

Санкт-Петербург

2019

СОДЕРЖАНИЕ

Введение	3
Постановка задачи	4
Обзор литературы	5
Глава 1. Обзор технологии распределенных реестров	6
1.1. Определение распределенного реестра	6
1.2. Классификация сетей распределенных реестров	7
1.3. Блокчейн	8
1.4. Смарт-контракты	8
Глава 2. Проблемы управления цепями поставок	9
2.1. Цепочка поставок	9
2.2. Причины внедрения технологий распределенного реестра	10
2.3. Существующие решения	11
Глава 3. Реализация	13
3.1. Архитектура системы	13
3.2. Выбор платформы	14
3.3. Реализация системы	16
3.4. Создание пользовательского интерфейса	18
Выводы	22
Заключения	23
Список литературы	24

Введение

Из-за бума, связанного с появившейся в 2009 году криптовалютой bitcoin, в основе которой лежит технология распределенных реестров, в данный момент наблюдается рост интереса в использовании этой технологии в других различных направлениях. [1]

Главная идея применения заключается в обеспечении прозрачных и надежных отношений между сторонами без централизованного контрольно-регулирующего посредника.

В настоящее время появляется все больше приложений на базе технологии распределенных реестров, подтверждая то, что существует множество форм его использования. Такие проекты имеют свои уникальные особенности, так как они направлены на решение различных нужд потребителя.

Постановка задачи

Прежде чем появиться на полках магазина, любой продукт проходит сложный путь, в котором обычно взаимодействуют между собой поставщики, производители, перевозчики, дистрибьюторы и ритейлеры. Управление такой цепочкой поставок может становиться все сложнее и сложнее при увеличении участников цепи. Такая проблема может быть решена применением технологии распределенных реестров, которая основана на смарт-контрактах, выполняющих условия соглашений автоматически.

Цель моей работы заключается в реализации такой системы, которая позволит совершать прозрачные сделки между участниками сети и отслеживать статус сделки.

Для достижения обозначенной цели необходимо выполнить следующие подзадачи:

1. Изучить предметную область
2. Выбрать платформу для работы
3. Реализовать систему товарообмена между участниками
4. Создать пользовательский интерфейс

Обзор литературы

Информация из [1-3] использовалась для изучения и понимания принципов работы технологии распределенных реестров, классификации сетей распределенных реестров и выделения их ключевых особенностей.

В источниках [4,5] описывается управление цепочками поставок, существующие проблемы в этой области, а также возможность их решения с применением технологии распределенных реестров. Источники [6-8] содержат найденные существующие коммерческие решения, реализованные с помощью технологии блокчейн.

Платформы, перечисленные в пунктах [9-11], рассматривались при поиске платформы, которая бы подошла под условия задачи и в которой не использовалась технология из-за выявленных в ней недостатков.

[14-16], [22] – это ссылки на официальную документацию и сайты программных продуктов, которые были рассмотрены или применены в рамках данной работы для достижения результата. В разработке также были применены перечисленные в пунктах [12-13], [19-21] языки и следующие методы [17-18], [23].

Глава 1. Обзор технологии распределенных реестров

1.1. Определение распределенного реестра

Под термином распределенный реестр подразумевается база данных, распределенная между несколькими сетевыми узлами или вычислительными устройствами. Каждый из этих узлов хранит полную копию реестра и может получать данные из других узлов. Обновление узлов происходит вне зависимости друг от друга.

Основная характеристика технологии распределенного реестра заключается в децентрализации, то есть отсутствии единого центрального управления. Все транзакции проверяются сетевыми узлами и записываются в свою распределенную базу данных. Затем проводится голосование за обновление реестра, для того, чтобы удостовериться в одобрении нового варианта большей частью узлов. Достижение консенсуса, то есть приход к согласию по поводу обновления реестра путем голосования, происходит автоматически с использованием алгоритмов консенсуса. С достижением консенсуса весь распределенный реестр обновляется, а утвержденная версия реестра сохраняется на каждом узле. [2]

Можно отметить несколько преимуществ технологии распределенных реестров, подталкивающих участников финансового рынка изучать, разрабатывать и внедрять данную технологию:

- отпадает необходимость в посреднике или центральном органе, контролирующем реестр
- повышенная прозрачность и стабильность ведения реестров сделок

- автоматическое выполнение условий соглашения при помощи смарт-контрактов
- увеличение устойчивости за счет распределенности и наличия большого количества копий данных

1.2. Классификация сетей распределенных реестров

Сети распределенных реестров можно разделить на три вида:

- 1) Открытые
- 2) Закрытые
- 3) Гибридные

- В открытых сетях распределенных реестров нет ограничений на доступ к сети. Личность ее участников не идентифицируется, и каждый желающий имеет возможность присоединиться к системе, читать или записывать в нее информацию. Отсутствуют какие-либо узлы, диктующие правила сети и имеющие над ней контроль. Самый известный пример использования открытых сетей распределенных реестров - криптовалютная сеть bitcoin.
- В закрытых сетях распределенных реестров существуют условия, согласно которым участникам разрешается присоединиться к сети. Подобные условия могут представлять собой какие-либо финансовые или юридические требования. В сетях такого типа все участники идентифицируемы, доступ к сети ограничен, а также существует какой-то ответственный узел, управляющий сетью и обладающий исключительными правами.

- Гибридные сети распределенных реестров совмещают свойства открытых и закрытых сетей.

1.3. Блокчейн

Технология блокчейн является одним из вариантов реализации сети на базе распределенных реестров. В ней информация о транзакциях между участниками хранится в виде цепочки связанных блоков. Важно понимать, что блокчейн является разновидностью распределенного реестра, но не каждый распределенный реестр использует последовательность блоков.

Одним из главных отличий блокчейна является то, что в нем можно только добавлять новые данные, то есть информацию в уже существующих блоках нельзя удалить или изменить. А каждый новый блок транзакций сначала должен признаться валидным остальными участниками сети и только после этого присоединиться к остальной цепочке.

1.4. Смарт-контракты

Смарт-контракт представляет собой электронный договор, который позволяет регулировать обмен активами между людьми или учреждениями без необходимости в стороннем посреднике. В нем прописаны права и обязательства, которые вступают в силу в автоматическом порядке при выполнении определенных условий.

У смарт-контрактов множество различных сфер применения, их можно использовать в страховании, кредитовании, передаче недвижимости, и даже в здравоохранении. [3]

Глава 2. Проблемы управления цепями поставок

2.1. Цепочка поставок

Цепочка поставок представляет собой все связи, которые вовлечены в планирование, поиск, производство, распределение и доставку каких-либо продуктов или услуг от места происхождения до места потребления. В настоящее время, подобные цепочки поставок, в зависимости от продукта, могут включать в себя сотни этапов, что делает процесс контроля очень сложным. [4] Также для оформления всех операций приходится работать с очень большим потоком документов, в которых бывают множество расхождений, связанных с человеческим фактором. Так например, если перевозчик и грузополучатель по-разному трактуют время доставки, может пострадать своевременность доставки. [5]

Сегодняшние цепочки поставок управляются субъектами, которые передают информацию следующему субъекту и запрашивают обновления. Электронная почта, телефонные звонки и неопределенность - это ежедневные подробности перемещения товаров. Передача информации такими способами грозит потерей целостности и подлинности данных.

Таким образом, система поставок в том виде, в каком она предстает сейчас, имеет множество недостатков:

- Отсутствие прозрачности в отношениях партнеров
- Различие данных у каждого участника системы
- Сложный процесс мониторинга этапов
- Возможность недобросовестного поведения партнеров

2.2. Причины внедрения технологий распределенного реестра

Используя технологию распределенного реестра можно решить изложенные выше проблемы, в первую очередь из-за того, что эта технология предназначена для установления доверия к данным и обеспечения поддержки единой версии истины у каждого участника сети.

Для плодотворного сотрудничества между участниками должно быть в первую очередь доверие. В открытых сетях распределенного реестра участники могут быть анонимными, что не дает оснований для полного доверия между ними. Тогда как в закрытой сети распределенных реестров полное доверие может быть достигнуто, так как доступ в такую сеть получают лишь те участники, которые проходят по определенным заранее критериям. Также в такой сети участники идентифицируемы.

Все обязательства между участниками прописываются в смарт-контрактах, еще больше укрепляя доверие. При выполнении договоренностей, он автоматически исполняет все указанные в нем условия, без постороннего вмешательства. Вдобавок они избавляют от бумажной волокиты, так как записываются в электронном виде.

Одним из главных принципов технологии распределенного реестра является его децентрализованность - то есть, никакая третья сторона не владеет сетью и не может изменять в ней данные для своей выгоды. Каждый участник сети будет видеть одну и ту же правдивую информацию, что облегчит обмен информацией между участниками.

Еще одним фактором является то, что с помощью технологии распределенного реестра можно решить проблемы мониторинга цепочки поставок, потому что она позволяет отслеживать транзакции в режиме реального времени.

Именно из-за всех перечисленных качеств, технология распределенного реестра имеет огромный потенциал для решения проблем, связанных с управлением цепочками поставок. Довольно универсальным будет реализация решения следующим образом: вовлеченные участники могут отмечать перемещение продукта и другие, важные для управления цепочкой поставок, данные с помощью транзакций.

2.3. Существующие решения

Несмотря на то, что технологии распределенных реестров совсем недавно начали развиваться, на сегодняшний день уже существует множество проектов, в основе которых лежит один из вариантов реализации такой технологии, а именно блокчейн. Он используется в решении задач связанных с перевозками, управлением запасами, проверкой подлинности, благотворительностью и многими другими. Ниже представлены наиболее известные проекты:

- Everledger Diamond Platform, разработанный компанией Everledger, регистрирует информацию об алмазах с помощью технологии блокчейн, которая объединяет всех участников отрасли, включая производителей, распространителей и покупателей. В реестре хранится информация о происхождении, цвете, огранке и других характеристиках алмаза, то есть всего, что необходимо для установления подлинности камня. [6]

- TradeLens является платформой, созданной с помощью технологии IBM Blockchain, решающей проблемы доверия в области перевозок. Она позволяет перевозчикам и владельцам грузов обмениваться информацией о транзакциях в режиме реального времени. [7]
- IBM Food Trust объединяет производителей, переработчиков, дистрибьюторов и розничных продавцов для регистрации данных о продовольственных товарах и предоставляет общий доступ к ним. Платформа использует технологию блокчейна для обеспечения прозрачности транзакций и отслеживаемости в цепочке поставок продуктов питания. [8]

У технологии блокчейн, которая используется в приведенных выше примерах, множество плюсов, но также есть и противоречивые моменты:

- У каждого участника есть копия каждой транзакции, а это очень много дублирующихся данных
- У каждого в сети есть доступ к информации, даже если она его не касается
- В блокчейне возникает проблема его масштабируемости

Глава 3. Реализация

3.1. Архитектура системы

Для реализации взаимодействия внутри цепочки поставок с применением технологии распределенных реестров таким образом, чтобы избежать обозначенных ранее недостатков, была разработана архитектура системы. Система товарообмена будет представлять собой базовую модель цепочки поставок, которую можно будет дополнить при необходимости, и будет содержать три типа участников:

- **Producer**, функция которого заключается в производстве некоего товара
- **Carrier**, который отвечает за перевозку товара в определенный магазин
- **Grocer**, участник, который получает товар и размещает его в магазине для дальнейшей продажи

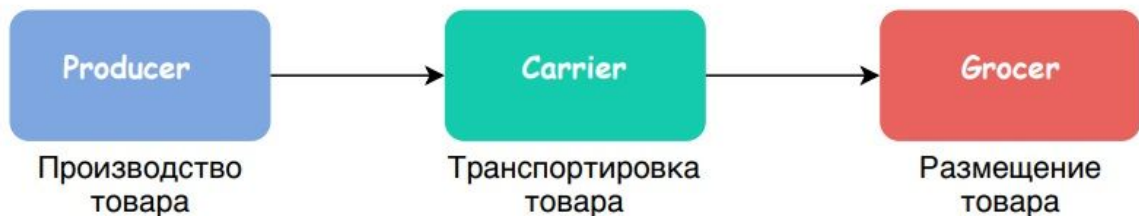


Рис. 1. Базовая модель цепочки поставок

Функционал программы должен включать в себя следующее:

- 1) Производитель, зайдя в систему, может создать заявку и выбрать участников, с которыми он хочет сотрудничать в своей цепочке поставок
- 2) После этого, заявка будет отправлена выбранным участникам, и если все будут удовлетворены условиями соглашения, то заявка сохранится

- 3) Перевозчик должен будет забрать товар у производителя и сменить статус товара с помощью приложения
- 4) Бакалейщик получает товар и отмечает в системе успешную доставку до магазина
- 5) С помощью сайта можно будет мониторить и контролировать статусы заказов

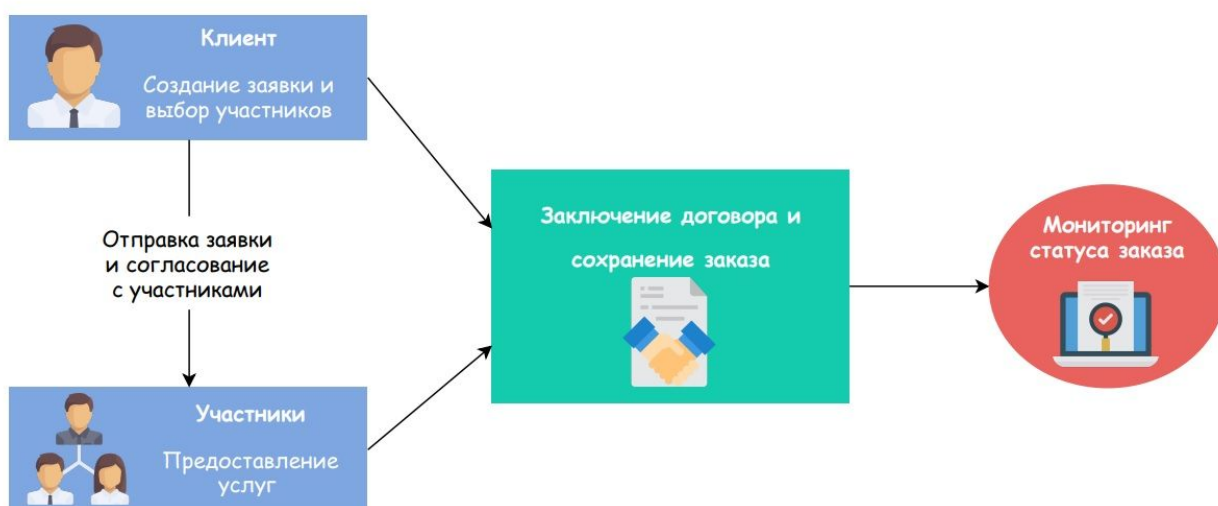


Рис. 2. Архитектура системы

Такая система будет реализовывать товарообмен между участниками сети и создаст фундамент, который можно будет использовать для более узкоспециализированного решения любого бизнес-проекта, связанного с управлением цепочками поставок.

3.2. Выбор платформы

Существует множество платформ, работающих на базе технологии распределенных реестров. Самыми популярными являются Ethereum [9],

Hyperledger Fabric [10], и многие другие. Мой выбор остановился на платформе Corda [11] в силу следующих причин:

- Проверка транзакций происходит исключительно между сторонами транзакции
- Закрытая сеть
- Поддержка различных механизмов консенсуса
- Контракты могут включать как машинный код, так и официальный юридический текст
- Используются языки Java [12] и Kotlin [13]

Corda представляет собой распределенную платформу с ограниченным доступом, и использующую JVM смарт-контракты. Несмотря на то, что Corda ориентирована в первую очередь на финансовый сектор, ее применение им не ограничено. Важным является то, что эта платформа не использует технологию блокчейн, вместо этого в ней применяются специальные нотариальные ноды, проверяющие транзакцию.

Транзакция записывается в реестр, в том случае, если она:

- Не содержит двойных трат
- Валидна согласно контракту
- Подписана необходимыми участниками

Если при использовании блокчейна каждый пользователь видит весь реестр, даже если он не имеет к нему никакого отношения, то в Corda записи не транслируются всем участникам, а доступны лишь тем участникам сети,

которые имеют к ним доступ. Таким образом, Corda подходит под решение заданной выше архитектуры.

Ключевыми понятиями в Corda являются:

- State Object: цифровой документ, который дает всю соответствующую информацию о соглашении, которое разделяется между сторонами
- Transactions: осуществляют переходы от существующего state object к новому state object
- Smart contract: гарантируют выполнение транзакций соответственно предварительно согласованным правилам

3.3. Реализация системы

При создании заявки, автоматически статусом заказа будет фирма производителя, и не будет изменяться до тех пор, пока транспортная компания не заберет товар и не изменит статус системы. Далее, товар должен быть доставлен в магазин, но перед сменой статуса должна осуществляться какая-либо проверка посылки на подлинность. Предположим, что для осуществления проверки того, что был доставлен действительно правильный товар, на каждой посылке есть идентификационный номер, который бакалейщик должен ввести в систему при получении заказа. Так как этот номер был присвоен при производстве, то при вводе в систему будет произведена автоматическая проверка на соответствие, и если номера совпадают, то товар оригинален, а заказ выполнен.

С помощью среды разработки IntelliJ IDEA [14] был реализован Corda проект. Также использовалась фреймворк Spring Boot [15] и система автоматической сборки Gradle [16].

В нашей системе, все взаимодействия, происходящие с товаром, фиксируются в состоянии - классе, в котором хранится вся информация и который проверяется контрактами. В Corda состояния неудаляемые и неизменяемые, что обеспечивает прозрачность системы, так как нельзя незаметно стереть или изменить информацию, записанную ранее. Состояние можно только заменить на новое, предварительно сделав старое неактуальным. [17] Итак, в главном состоянии хранится информация о производителе, перевозчике, бакалейщике, идентификационном номере и текущем состоянии заказа. При переходе товара от одного участника цепи к другому новая информация записывается в реестр, а старые данные сохраняются в Vault - хранилище данных, который содержит всю информацию о неактуальных состояниях и к которому можно легко обратиться. Изменения в системе происходят тогда, когда какой-либо узел запускает функцию-поток, с целью обновить реестр. Обычно с их помощью выполняются и проверяются транзакции, а также собираются подписи участвующих в изменении узлов. [18] В проекте, процесс товарообмена реализован на базе функций-потоков, через которые обновляется статус состояния товара. Также, когда бакалейщик вводит в систему идентификационный номер с целью проверить товар, функция сравнивает значения введенного идентификационного номера со значением из хранилища Vault, и при совпадении доставка становится выполненной.

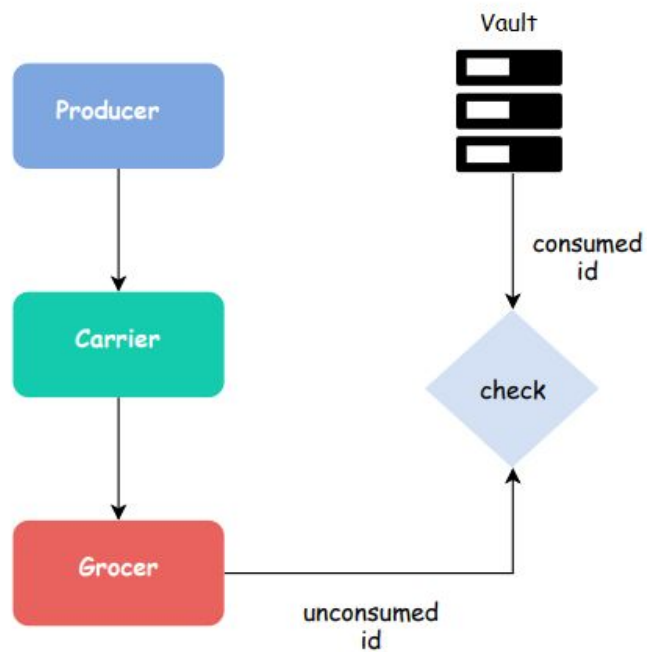


Рис. 3. Схема проверки идентификационного номера

3.4. Создание пользовательского интерфейса

Данный этап включает в себя создание привлекательного пользовательского интерфейса. Фронтенд был реализован с помощью языков HTML [19], CSS [20] и Javascript [21], а также был использован инструмент для создания пользовательских интерфейсов React[22]. Его главное преимущество в том, что он значительно облегчает создание интерфейсов благодаря разбиению страницы на небольшие фрагменты, называемые компонентами. Также он позволяет сразу же видеть вносимые изменения, причем обновляется не полностью вся страница, а лишь та часть, в которой произошли изменения.

Клиентская часть была разделена на следующие компоненты:

- MakeOrder: Предоставляет возможность добавить новый заказ и выбрать остальных участников

- Orders: Визуализация всех заказов текущего пользователя
- OrderBox: Отображает все сведения отдельного заказа: его номер, дату заключения контракта, текущий статус и выполняющих заказ участников
- Header: Выполняет роль навигации на странице и позволяет перейти от страницы всех заказов к созданию нового и обратно

Во-первых, нужно было реализовать удобный интерфейс для пользователя при мониторинге и отслеживании статуса заказов. При переходе на страницу заказов автоматически отправляется fetch запрос [23] и выводятся все заказы, связанные с пользователем, который вошел в систему. В каждом заказе можно увидеть участников, выполняющих этот заказ, а также графу текущий статус заказа, в которой отображается тот участник, который выполняет заказ в данный момент.

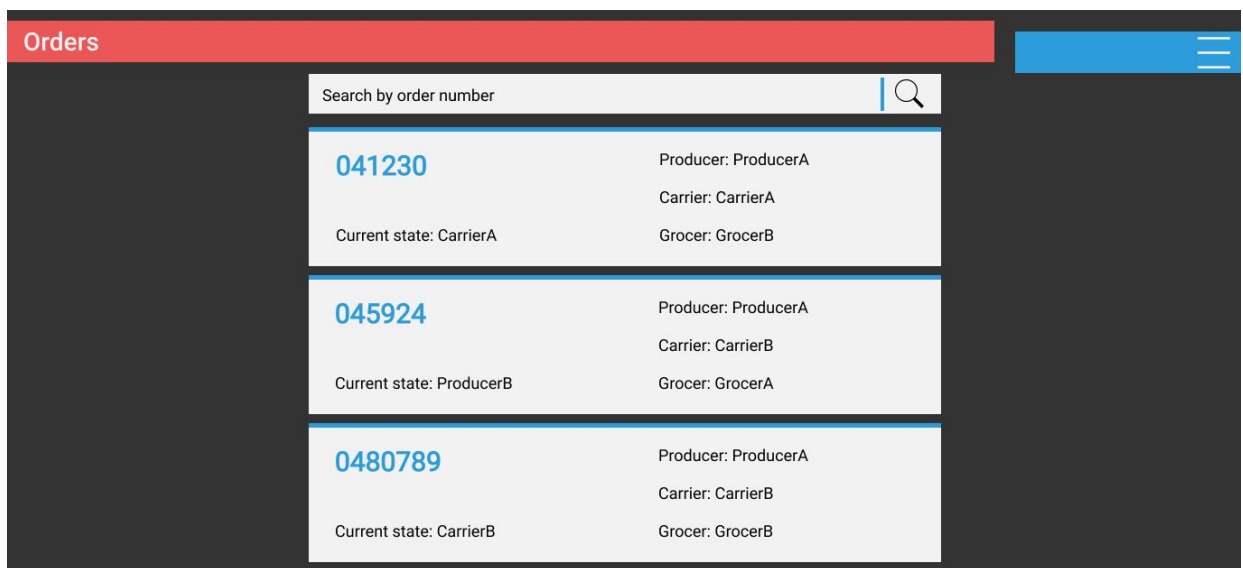


Рис. 4. Все заказы, связанные с ProducerA

Так как у каждого пользователя выполняются различные действия, связанные с товаром, для каждого из них были разработаны свои формы заполнения для работы с системой. Например, производитель с помощью

интерфейса может легко отправить заявку и выбрать участников своей цепочки. Так как у нас закрытая сеть, набор возможных участников известен заранее. После отправки новой заявки и согласования ее с другими участниками, ее можно будет увидеть со всеми остальными заказами на соответствующей странице. Перевозчик может поменять статус заказа, чтобы отметить, что товар был успешно принят у производителя и сейчас находится в пути. Бакалейщик же, при получении посылки, вводит уникальный номер в систему, и если все верно, заказ меняет статус и становится выполненным.

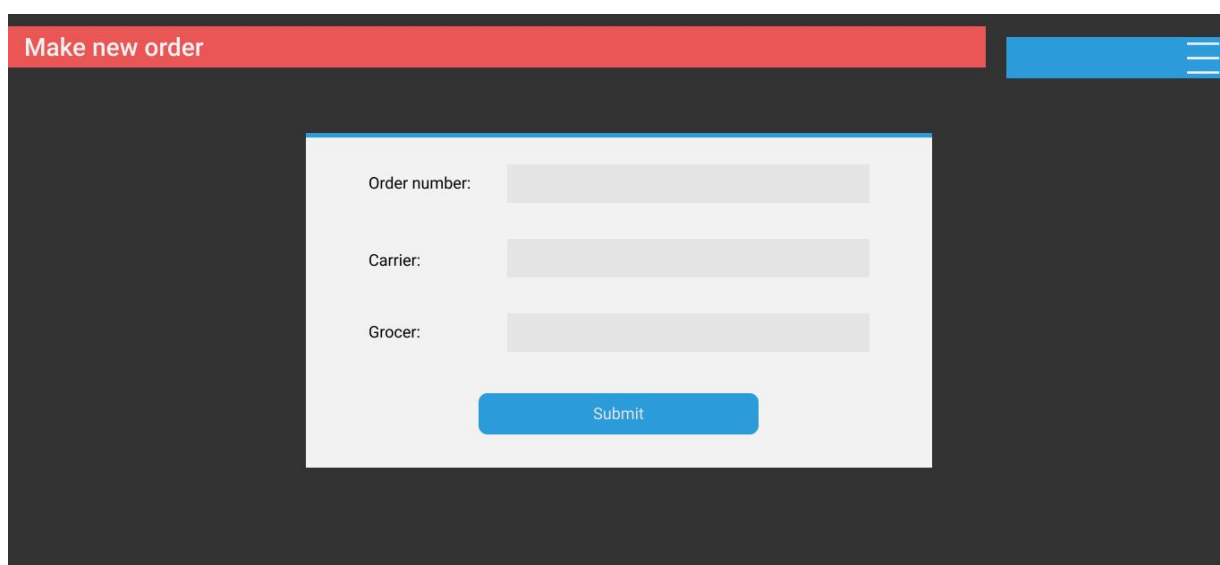
The image shows a web interface for creating a new order. At the top, there is a red header bar with the text "Make new order" in white. To the right of the header is a blue navigation menu icon. The main content area is dark gray. In the center, there is a white form box with a blue border. The form contains three input fields: "Order number:", "Carrier:", and "Grocer:". Below these fields is a blue "Submit" button.

Рис. 5. Оформление нового заказа через производителя

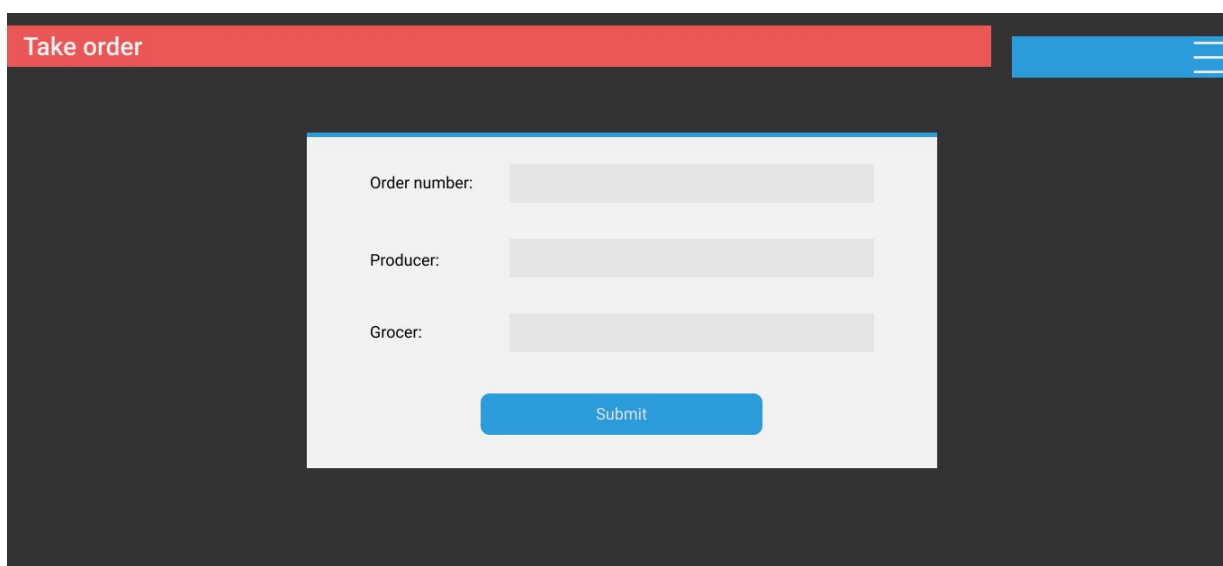
The image shows a web interface for taking an order. At the top, there is a red header bar with the text "Take order" in white. To the right of the header is a blue navigation menu icon. The main content area is dark gray. In the center, there is a white form box with a blue border. The form contains three input fields: "Order number:", "Producer:", and "Grocer:". Below these fields is a blue "Submit" button.

Рис. 6. Форма перевозчика

The image shows a web interface for checking an order. At the top left, there is a red header bar with the text "Check order". To the right of this bar is a blue navigation menu icon. The main content area is dark grey. In the center, there is a white rounded rectangle containing a form. The form has three input fields: "Order number:", "Producer:", and "Id-number:". Each field is followed by a light grey rectangular input box. Below these fields is a blue button with the text "Submit".

Рис. 7. Проверка идентификационного номера для бакалейщика

В реализованной системе использовался идентификационный номер для проверки правильности доставленного товара. В дальнейшей разработке его можно заменить на штрих-код или QR-код, чтобы была возможность проверять товар с помощью сканера телефона. В данной работе это не было реализовано из-за необходимости дополнительных трудовых и денежных затрат, поэтому реализован ручной набор идентификационного номера.

Выводы

В рамках данной работы был проведен анализ сегодняшних проблем в управлении цепочками поставок. Были рассмотрены существующие подходы и готовые решения перечисленных проблем с помощью разновидности технологии распределенного реестра, а именно блокчейн технологии, и был предложен вариант реализации системы товарообмена с использованием платформы Corda, не использующей последовательность блоков. В результате было разработано приложение, представляющее собой базовую модель реализующую товарообмен, с возможностью мониторинга заказов. Для хорошего восприятия также был создан понятный пользовательский интерфейс.

Заключения

С целью предложить решение для товарообмена в сфере управления цепочками поставок было разработано веб-приложение с использованием технологии распределенного реестра, которое позволяет обмениваться активами с остальными участниками сети и следить за изменением статуса товара, а самое главное поддерживает доверие между участниками, за счет прозрачности операций. При необходимости, на основе созданной системы можно создать любой проект, связанный с цепочками поставок и их управлением. После конкретизации области использования может быть необходимо отредактировать условия работы смарт-контрактов, добавить дополнительных участников цепи и сделать другие преобразования, связанные со спецификой выпускаемого продукта.

Список литературы

1. Развитие технологии распределенных реестров URL: https://www.cbr.ru/Content/Document/File/36007/reestr_survey.pdf (дата обращения: 30.05.2019)
2. Чем различаются блокчейн и распределенный реестр URL: <https://ethclassic.ru/2018/04/11/chem-otlichayutsya-blokchejn-i-raspredelennyy-reestr> (дата обращения: 30.05.2019)
3. Smart Contracts: The Blockchain Technology That Will Replace Lawyers URL: <https://blockgeeks.com/guides/smart-contracts> (дата обращения: 30.05.2019)
4. В. П. Куприяновский, С. А. Синягов, А. А. Климов, А. В. Петров, Д. Е. Намиот Цифровые цепи поставок и технологии на базе блокчейн в совместной экономике // International Journal of Open Information Technologies. 2017. №8.
5. Блокчейн в логистике: как технология помогает цепи поставок? URL: <https://www.artlogics.ru/blog/blockchain-v-logistike> (дата обращения: 30.05.2019)
6. Everledger Diamond Platform URL: <https://www.everledger.io> (дата обращения: 30.05.2019)
7. TradeLens URL: <https://www.tradelens.com> (дата обращения: 30.05.2019)
8. IBM Food Trust: adding trust and transparency to our food URL: <https://www.ibm.com/blockchain/solutions/food-trust> (дата обращения: 30.05.2019)

9. Ethereum URL: <https://www.ethereum.org> (дата обращения: 30.05.2019)
10. Hyperledger Fabric URL: <https://www.hyperledger.org/projects/fabric> (дата обращения: 30.05.2019)
11. Corda URL: https://docs.corda.net/_static/corda-introductory-whitepaper.pdf (дата обращения: 30.05.2019)
12. Java URL: <https://www.oracle.com/technetwork/java/index.html> (дата обращения: 30.05.2019)
13. Kotlin URL: <https://kotlinlang.org> (дата обращения: 30.05.2019)
14. IntelliJ IDEA URL: <https://www.jetbrains.com/idea> (дата обращения: 30.05.2019)
15. Spring Boot URL: <https://spring.io/projects/spring-boot> (дата обращения: 30.05.2019)
16. Gradle Building Tool URL: <https://gradle.org> (дата обращения: 30.05.2019)
17. Vault URL: <https://docs.corda.net/vault.html> (дата обращения: 30.05.2019)
18. Flows URL: <https://docs.corda.net/key-concepts-flows.html> (дата обращения: 30.05.2019)
19. HTML URL: <https://html5book.ru/html-tags> (дата обращения: 30.05.2019)
20. CSS URL: <https://html5book.ru/osnovy-css> (дата обращения: 30.05.2019)
21. JavaScript URL: <https://www.javascript.com> (дата обращения: 30.05.2019)
22. React URL: <https://ru.reactjs.org> (дата обращения: 30.05.2019)
23. Метод fetch URL: <https://learn.javascript.ru/fetch> (дата обращения: 30.05.2019)