УДК 004.9

АРХИТЕКТУРНЫЙ ПОДХОД К ОБЕСПЕЧЕНИЮ СИНХРОНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ЦИФРОВЫХ ФИНАНСОВЫХ АКТИВОВ

Гаджиев А.М.

МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: yudaarsen.ca@gmail.com

Цифровые финансовые активы находят широкое применение за счет удобства использования, большей доступности для инвесторов и сокращения издержек при размещения эмитентами. Тем не менее, как и традиционные финансовые инструменты, цифровые финансовые активы подвержены техническим и инфраструктурным рискам. Снизить последствия в случае реализации данных рисков возможно путем реализации распределенных между несколькими информационными системами цифровых финансовых активов, что, в свою очередь, потребует обеспечения синхронизации между платформами. Целью работы является разработка архитектурного подхода к обеспечению синхронизации распределенных цифровых финансовых активов.

Ключевые слова: ЦФА, цифровые финансовые активы, Ethereum, смарт-контракт, HyperLedger, EIP, блокчейн, шаблоны проектирования, проектирование.

ARCHITECTURAL APPROACH TO ENSURING SYNCHRONIZATION OF DISTRIBUTED DIGITAL FINANCIAL ASSETS

Gadzhiev A.M.

MIREA - Russian Technological University, 119454, Moscow, 78 Vernadskogo Avenue, Russia, e-mail: yudaarsen.ca@gmail.com.ru

Digital financial assets are widely used because of their ease of use, greater accessibility to investors and reduced costs for issuers. Nevertheless, like traditional financial instruments, digital financial assets are subject to technical and infrastructural risks. It is possible to reduce the consequences in case these risks materialize by implementing digital financial assets distributed between several information systems, which, in turn, will require ensuring synchronization between platforms. The aim of the work is to develop an architectural approach to ensure synchronization of distributed digital financial assets.

Keywords: DFA, digital financial assets, Ethereum, smart-contract, HyperLedger, EIP, blockchain, design patterns, design.

Введение

Цифровые финансовые активы – это новый инструмент финансового рынка, появившийся в Российской Федерации в 2020 году. Цифровые финансовые активы реализуются на основе технологии распределенного реестра и существенно упрощают финансовые операции с активами [1]. Цифровизация реальных активов делает инвестиции доступнее, а финансовые операции проще и быстрее. Доступность инвестиций в цифровые финансовые активы обуславливается отсутствием привязки к

присущим конкретному виду актива единицам измерения при совершении сделок. Например, в отличие от традиционных инструментов покупки драгоценных металлов (на бирже или с помощью обезличенного металлического счета), минимальный лот которых, характеризуется весом конкретного металла, цифровая форма данных активов позволяет реализовать большую фрагментарность, что в свою очередь обеспечивает возможность инвесторам совершать сделки с объемами, меньшими, чем минимальный лот в традиционном варианте.

Цифровой формат активов открывает возможность к обеспечению интероперабельности информационных систем, в которых осуществляется учет, выпуск и оборот цифровых финансовых активов, а также и корпоративных информационных систем, что повышает эффективность выполнения операций. Ещё одним весомым аргументом для эмитентов ценных бумаг является снижение издержек, связанных с выпуском цифровых финансовых активов, за счет отсутствия необходимости в предоставлении документации на бумажных носителях. Преимуществом технологии цифровых финансовых активов является и высокая надежность, обеспеченная использованием технологии распределенного реестра.

Цифровые финансовые активы выпускаются, учитываются и обращаются в рамках информационной системы, что создает зависимость инвесторов и эмитентов от технических характеристик качества конкретной платформы. Российские инвесторы подвержены существенным инфраструктурным ограничениям на рынке ценных бумаг иностранных эмитентов, из-за которых стало невозможным выполнение ряда операций с ценными бумагами [2]. Кроме того, участники финансового рынка также подвержены инфраструктурным рискам при выпуске цифровых финансовых активов, связанных с решением регулятора о прекращении обращения цифровых финансовых активов [3]. Купировать инфраструктурный риск возможно путем реализации распределенных цифровых финансовых активов, под которыми, в настоящей работе, подразумеваются цифровые финансовые активы, синхронизированные между разными информационными системами.

Механизм достижения консенсуса

Распределенный реестр – это реестр, используемый множеством распределенных узлов, и который синхронизируется между данными узлами в соответствии с механизмом достижения консенсуса [4]. Механизм достижения консенсуса является одним из основных элементов распределенного реестра и позволяет узлам пиринговой сети достигать соглашения о текущем состоянии реестра по установленным правилам. Каждый консенсус разработан с допущением, что честных узлов, заинтересованных в соблюдении установленных правил, всегда большинство. Перечень наиболее распространенных консенснусных механизмов представлен в таблице 1.

Различные механизмы достижения консенсуса могут использоваться как в публичных, так и в закрытых блокчейн-платформах, в зависимости от цели использования распределенного реестра. Так, например, консенсус PoW при корпоративном использовании, в рамках которого между участниками уже было выстроено взаимодействие, является избыточным с точки зрения использования вычислительных ресурсов. В настоящее время в ряде существующих информационных системах, в которых осуществляется учет, выпуск и обращение цифровых финансовых активов, решения строятся на базе технологии HyperLedger Fabric [5, 6], которая

реализует механизм достижения консенсуса Byzantine Fault-Tolerant [7]. Разнообразие механизмов достижения консенсуса позволяет бизнесу и государству гибко настраивать блокчейн-платформу для решения финансовых и управленческих задач.

Таблица 1. Механизмы достижения консенсуса

Название механизма достижения консенсуса	**Описание**	**Примеры реализующих платформ**
Доказательство работы (*Proof of Work, PoW*)	Механизм основан на задаче, которая может быть решена только угадыванием. Например, когда на одном из узлов сформирован новый блок с проверенными транзакциями необходимо подобрать такое числовое значение (nonce в Bitcoin), чтобы итоговый хэш блока удовлетворял определенному формату (например, чтобы в начале было заданное количество нолей). Каждый узел сети случайным образом подбирает данное значение, чтобы получить хэш блока, удовлетворяющий требованиям. Таким образом, узлу необходимо тратить существенные вычислительные ресурсы с целью решения данной задачи быстрее остальных участников блокчейн-сети.	Bitcoin Dogecoin Litecoin
Доказательство доли (*Proof of Stake, PoS*)	PoS в отличие от PoW требует меньше вычислительных ресурсов. Принцип консенсуса построен на наличии специальных узлов (валидаторов), которые сделали определенный вклад. Среди таких узлов выбирается создатель очередного блока. Если выбранный создатель попытается добавить некорректный блок, то его вклад аннулируется. За каждый сформированный блок валидатор получает комиссию.	Ethereum Cardano
Делегированное доказательство доли (*Delegated Proof of Stake, DPoS*)	В DPoS валидация транзакций и защищенность сети обеспечиваются делегатами. Делегаты выбираются среди узлов путем голосования. Чем больше токенов имеет узел, тем больший вес голоса у него есть. В отличие от PoW, который требует существенных вычислительных ресурсов, и PoS, требующий весомый вклад для возможности создания блоков, в DPoS разрешено голосовать за узел, который будет генерировать новый блок и награждать только лучшие узлы.	Lisk Ark
Практическая византийская отказоустойчивость (*Practical Byzantine Fault Tolerance, PBFT*)	Принцип работы алгоритма состоит из двух основных аспектов: – узел, получивший транзакцию, осуществляется рассылку всем узлам; – каждый узел, получивший данные от других узлов, сверяет их и принимает транзакцию, если получено больше 2/3 голосов (подтверждений).	HyperLedger Fabric

Смарт-контракт

Смарт-контракт – это запрограммированное приложение, которое хранится в распределенном реестре. Первой широко известной платформой, которая позволила реализовывать смарт-контракты, стала платформа Ethereum. Для достижения консенсуса каждый узел распределенного реестра должен иметь специальное программное обеспечение, чтобы выполнять программный код, а также сам алгоритм смарт-контракта должен быть детерминированным, чтобы возвращать один и тот же результат на каждом узле. Детерминированность смарт-контрактов в публичных блокчейн-сетях достигается за счет использования специальных языков программирования, как например Solidity. Архитектура HyperLedger Fabric позволяет использовать языки общего назначения для реализации смарт-контрактов. В терминологии HyperLedger Fabric смарт-контракт называется чейнкодом (chaincode) и имеет ряд особенностей при развертывании

программного кода в блокчейн-сети. На рисунке 1 представлена архитектура HyperLedger Fabric.

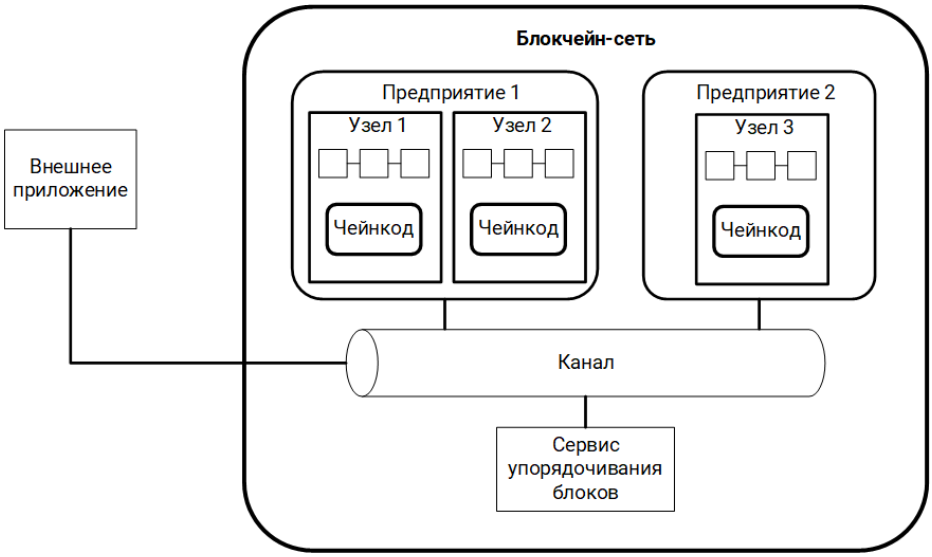

Рис.1 - Архитектура HyperLedger Fabric

Архитектура имеет следующие основные компоненты:

–внешнее приложение, инициирующее транзакции;

–узлы, развернутые в рамках предприятий (в контексте информационной системы цифровых финансовых активов, предприятия, которые разворачивают подтверждающие узлы, являются валидаторами);

–канал, обеспечивающий доступность блокчейн-сети авторизованным клиентам и узлам;

–сервис упорядочивая блоков, регулирующий последовательность сформированных блоков в блокчейн-сети.

Внешние приложения и узлы взаимодействуют в рамках канала, что обеспечивает конфиденциальность совершаемых транзакций в соответствии с нормативно-правовыми актами в области цифровых финансовых активов. Оператор информационной системы, то есть юридическое лицо, обеспечивающее функционирование информационной системы цифровых финансовых активов, регулирует доступ клиентов платформы к тому или иному каналу. Каждый узел при этом хранит блокчейн каждого канала, к которому обеспечен доступ, а также хранит чейнкод, который используется для подтверждений транзакций. Сервис упорядочивая блоков предназначен для установки единого порядка блоков в блокчейн-сети. Новый блок, сформированный сервисом упорядочивания блоков, рассылается всем узлам и добавляется в каждый локальный реестр.

Неотъемлемым свойством смарт-контрактов является их неизменность с момента добавления в распределенный реестр. Такая особенность является как преимуществом, так и недостатком. С одной стороны, участники взаимодействия могут быть уверены, что логика, описанная в смарт-контракте, не изменится, но с другой стороны, ошибки, заложенные в смарт-контракте, нельзя исправить. Кроме того, неизменяемость смарт-контрактов не позволяет дорабатывать существующую логику для реализации новых бизнес-функций или дополнительных механизмов синхронизации со внешними

платформами.

Архитектура HyperLedger Fabric позволяет хранить чейнкод на выбранных узлах блокчейн-сети, а также осуществлять обновление в рамках жизненного цикла чейнкода [8]. Технология также позволяет использовать языки программирования общего назначения для реализации логики чейнкода и выполнения запросов ко внешним сервисам. Несмотря на реализованный механизм обновления чейнкода, остается возможным нарушение синхронизации распределенных между несколькими платформами цифровых финансовых активов при изменении механизмов синхронизации. С целью достижения консенсуса, информация из удаленных ресурсов должна в первую очередь быть добавлена в блокчейн для дальнейшего использования, для этого создается отдельный чейнкод, выполняющий удаленные запросы. Цифровые финансовые активы, обращающиеся на разных платформах, должны своевременно синхронизировать свое состояние. В случае изменения механизма синхронизации или необходимости в переключении на дополнительный способ обмена данными между платформами, потребуются временные затраты для обновления чейнкода на всех узлах сети, что, в свою очередь, может привести к рассинхронизации состояний между платформами. Решение описанной проблемы может быть достигнуто за счет применения архитектурного подхода при проектировании цифровых финансовых активов, который позволит своевременно изменить механизм синхронизации без необходимости повторного развертывания чейнкода на узлах блокчейн-сети.

Анализ архитектурных подходов к проектированию обновляемых смарт-контрактов

В рамках платформ, на которых логика и состояние смарт-контрактов хранится непосредственно в блокчейне, могут использоваться следующие подходы к созданию обновляемых смарт-контрактов [**Ошибка! Источник ссылки не найден.**]:

1. Миграция контракта, предполагающее создание нового смарт-контракта в распределенном реестре и переносе состояния со старой версии контракта (рисунок 2). Подход имеет ряд недостатков:

–все смарт-контракты, которые обращались к обновляемому контракту, должны быть также обновлены, чтобы ссылаться на актуальную версию;

–клиенты (внешние приложения) должны быть оповещены об изменении версии контракта и самостоятельно начать использовать новый адрес.

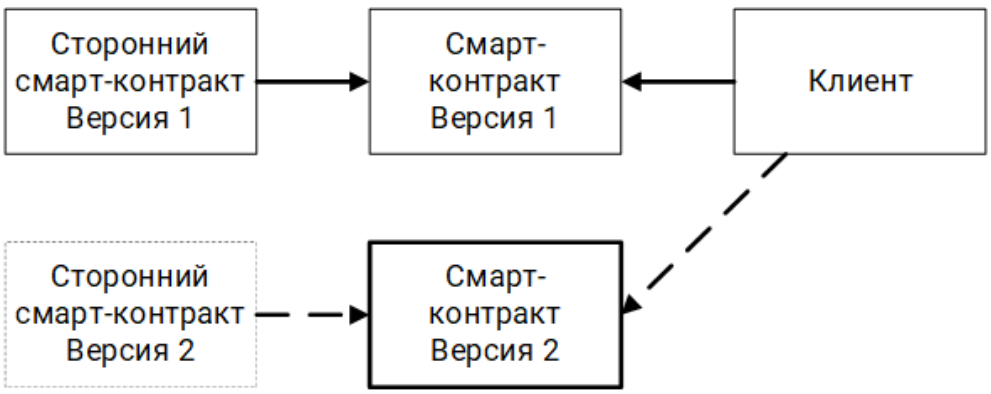

Рис. 1 - Миграция смарт-контракта

2. Отделение данных от бизнес-логики, путем выделения двух отдельных смарт-контрактов: логический смарт-контракт и смарт-контракт хранилище (рисунок. 2). Подход позволяет избавиться от необходимости в миграции состояния предыдущей версии смарт-контракта. Однако, недостатком является необходимость в обновлении адреса логического смарт-контракта у клиентов и сторонних контрактов в момент развертывания новой версии в распределенном реестре.

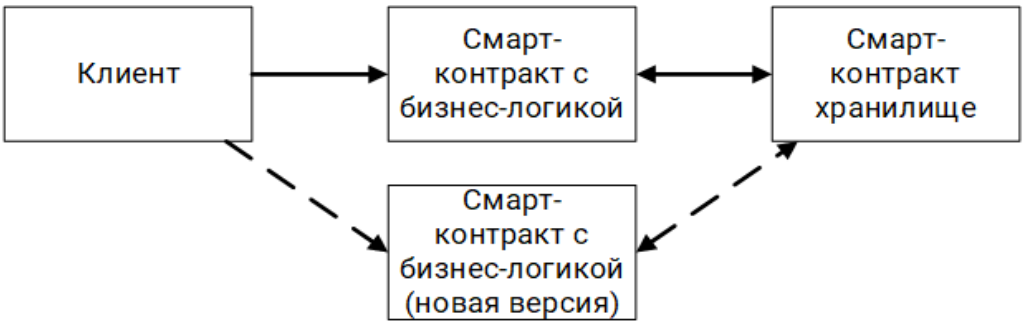

Рис. 2 - Отделение данных от бизнес-логики

3. Использование прокси-контракта, который хранит данные и осуществляет вызовы по адресу логического смарт-контракта [**Ошибка! Источник ссылки не найден.**]. При использовании такого подхода, запрос первоначально попадает на прокси-контракт, который хранит данные, но не содержит бизнес-логики, затем запрос перенаправляется на логический смарт-контракт по актуальному адресу (рисунок Рис. 3). Подход устраняет необходимость в изменении адреса новой версии контракта на клиентской стороне и сторонних смарт-контрактах, а также отсутствует необходимость в миграции состояния, поскольку все необходимые данные хранятся в прокси-контракте, с которым взаимодействуют клиенты.

Рис. 3 - Использование прокси смарт-контракта

Архитектура HyperLedger Fabric реализует разделение состояния и бизнес-логики чейнкода, а также позволяет развертывать и обновлять чейнкод на отдельных узлах блокчейн-сети. Таким образом, рассмотренные подходы не имеют целесообразности применения в блокчейн-сети на базе HyperLedger Fabric, поскольку бизнес-логика чейнкода может быть обновлена на узлах инструментальными средствами, а также архитектурно не предполагается использование адресов для работы с чейнкодом. Тем не менее, для задачи синхронизации распределенных между платформами цифровых финансовых активов, необходимо быстрое переключение механизма синхронизации, которое невозможно осуществить инструментальными средствами из-за временных

затрат на обновление чейнкода на распределенных узлах блокчейн-сети.

Шаблон проектирования «Стратегия» позволяет реализовать быстрое и своевременное переключение бизнес-логики смарт-контракта [**Ошибка! Источник ссылки не найден.**]. Подход похож на реализацию прокси-контракта с тем отличием, что контракт, с которым взаимодействуют клиенты, содержит основную бизнес-логику и предоставляет интерфейс для взаимодействия с другими логическими контрактами (рисунок 5Рис. 4). Применение данного шаблона проектирования позволяет реализовывать небольшие доработки без необходимости обновления всего логического контракта и осуществлять быстрое переключение между реализациями тех или иных функций.

Рис. 4 - Шаблон проектирования «Стратегия»

Шаблон проектирования «Алмаз» улучшает подход с использованием прокси-контракта добавлением возможности вызова функций из разных логических контрактов [**Ошибка! Источник ссылки не найден.**]. Подход позволяет реализовывать гибкую логику смарт-контракта, упрощает доработку отдельных небольших функций без необходимости обновления всего логического контракта, а также позволяет преодолевать ограничения по размерам смарт-контрактов на платформах, в которых программный код хранится в блокчейне (например, для платформы Ethereum максимальный размер контракта составляет 24 Кб).

Решение задачи синхронизации распределенных цифровых финансовых активов может быть достигнуто с использованием шаблонов проектирования «Стратегия» и «Алмаз». Тем не менее, шаблон проектирования «Алмаз» является избыточным, поскольку, в контексте технологии HyperLedger, нет необходимости в преодолении ограничений максимального размера чейнкода. В свою очередь, шаблон проектирования «Стратегия» выделяет основной интерфейс, используемый цифровыми финансовыми активами для синхронизации, а также вызывает выполнение определенных функций из логических контрактов, переключение между которыми может быть обеспечено выполнением одной транзакции. Предлагаемый архитектурный подход к обеспечению синхронизации распределенных цифровых финансовых активов представлен на рис. 6.

На рис. 6 представлено направление синхронизации от информационной системы 1 в информационную систему 2. В качестве триггеров синхронизации выступают определенные события цифрового финансового актива (например, смена владельца). Выделены следующие основные компоненты:

1. Информационная система – система, в которой осуществляется выпуск, учет и

обращение цифровых финансовых активов.

2. Система информационная обмена – система, обрабатывающая поступающие запросы от одной информационной системы и инициирующая транзакцию в другой информационной системе. Компонент может быть представлен, в том числе, в виде специального агента в рамках информационной системы.

3. Интерфейс – чейнкод, содержащий основные функции взаимодействия и реализацию базовой логики. Компонент сохраняет название актуального чейнкода, с помощью которого осуществляется формирование или обработка запроса синхронизации.

4. Логика формирования запроса/Обработчик запроса – чейнкод, содержащий конкретную реализацию логики синхронизации цифровых финансовых активов.

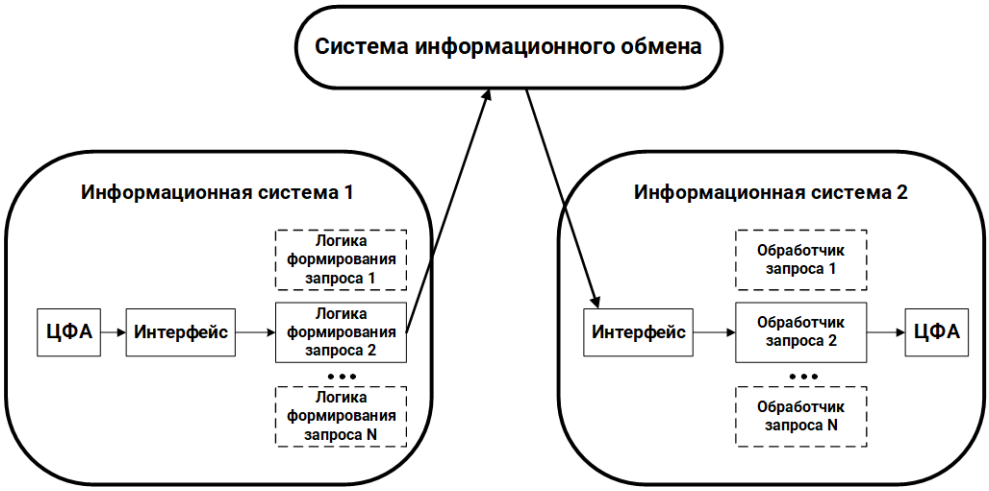

Рис. 6. -Архитектурный подход к обеспечению синхронизации распределенных цифровых финансовых активов

Предложенный архитектурный подход позволяет синхронизировать состояния цифровых финансовых активов между разными информационными системами.

Заключение

Цифровые финансовые активы, как и традиционные финансовые инструменты, подвержены инфраструктурным и техническим рискам в силу того, что выпуск, учет и обращение осуществляется в рамках одной информационной системы.

Инвесторы и эмитенты цифровых финансовых активов заинтересованы в обеспечении доступности инструмента и возможности выполнения необходимых транзакций.

Снижение последствий от реализации данных рисков достигается путем использования распределенных цифровых финансовых активов, которые нуждаются в своевременной синхронизации между платформами для поддержания согласованного состояния.

Результат анализа архитектурных подходов к проектированию обновляемых смарт-контрактов показал, что шаблон проектирования «Стратегия» является необходимым и достаточным для решения задачи синхронизации.

На основании данного шаблона проектирования предложен архитектурный подход, который может применяться для обеспечения синхронизации распределенных цифровых финансовых активов.

Список литературы

1. Федеральный закон от 31.07.2020 № 259-ФЗ «О цифровых финансовых активах, цифровой валюте и о внесении изменений в отдельные законодательные акты Российской Федерации» (Дата обращения 18.04.2023)

2. ПАО СПБ Биржа. Уведомление о рисках, связанных с приобретением ценных бумаг иностранных эмитентов. Режим доступа: https://spbexchange.ru/ru/stocks/inostrannye/uv_risk/ (Дата обращения: 18.04.2023)

3. Указание Банка России № 6336-У «О порядке применения Банком России к оператору информационной системы, в которой осуществляется выпуск цифровых финансовых активов, мер, предусмотренных частью 18 статьи 5 Федерального закона от 31 июля 2020 года № 259-ФЗ «О цифровых финансовых активах, цифровой валюте и о внесении изменений в отдельные законодательные акты Российской Федерации», и к оператору обмена цифровых финансовых активов мер, предусмотренных частью 17 статьи 10 Федерального закона от 31 июля 2020 года № 259-ФЗ «О цифровых финансовых активах, цифровой валюте и о внесении изменений в отдельные законодательные акты Российской Федерации»

4. ISO 22739:2020. Blockchain and distributed ledger technologies – Vocabulary. (Дата обращения: 18.04.2023)

5. Atomyze. Режим доступа: https://atomyze.ru/press-reliz (Дата обращения 19.04.2023)

6. Правила информационной системы Сбера, в которой осуществляется выпуск цифровых финансовых активов. Режим доступа: http://www.sberbank.ru/common/img/uploaded/legal/docs/digital-assets/pravila_inf_sistemy.pdf (Дата обращения 19.04.2023)

7. HyperLedger Fabric. Docs. Режим доступа: https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html (Дата обращения 19.04.2023)

8. HyperLedger Fabric. Fabric chaincode lifecycle. Режим доступа: https://hyperledger-fabric.readthedocs.io/en/latest/chaincode_lifecycle.html (Дата обращения: 20.04.2023)

9. Ethereum Docs. Upgrading smart contracts. Режим доступа: https://ethereum.org/en/developers/docs/smart-contracts/upgrading/ (Дата обращения: 20.04.2023)

10. Ethereum Improvement Proposals. ERC-1967: Proxy Storage Slots. Режим доступа: https://eips.ethereum.org/EIPS/eip-1967 (Дата обращения: 20.04.2023)

11. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. Паттерны объектно-ориентированного проектирования / Пер. с англ.: А. Слинкин. — СПб.: Питер, 2021. — 448 с. — ISBN 978-5-4461-1595-2.

12. Ethereum Improvement Proposals. ERC-2535: Diamonds, Multi-Facet Proxy. Режим доступа: https://eips.ethereum.org/EIPS/eip-2535 (Дата обращения: 20.04.2023)

References

1. The federal law of the Russian Federation of 31.07.2020 № 259-FZ «On digital financial assets, digital currency and on amendments to certain legislative acts of the Russian Federation» (Accessed 18.04.2023)

2. St. Petersburg Stock Exchange PJSC. Notification of risks associated with the acquisition of securities of foreign issuers. Access mode: https://spbexchange.ru/ru/stocks/inostrannye/uv_risk/ (Accessed 18.04.2023)

3. Instruction of the Bank of Russia № 6336-U "On the order of application by the Bank of Russia to the operator of the information system, in which the issue of digital financial assets, measures provided by part 18 of Article 5 of the Federal Law of July 31, 2020 № 259-FZ" On digital financial assets, digital currency and on amendments to certain legislative acts of the Russian Federation, and to the exchange operator of digital financial assets measures provided for by part 17 of Article 10 of Federal Law of July 31, 2020 № 259-FZ "On digital financial assets, digital currency and digital currency.

4. ISO 22739:2020. Blockchain and distributed ledger technologies – Vocabulary. (Accessed 18.04.2023)

5. Atomyze. Access mode: https://atomyze.ru/press-reliz (Accessed 19.04.2023)

6. Rules of the information system of Sbera, in which digital financial assets are issued. Access mode: http://www.sberbank.ru/common/img/uploaded/legal/docs/digital-assets/pravila_inf_sistemy.pdf (Accesed 19.04.2023)

7. HyperLedger Fabric. Docs. Access mode: https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html (Accessed 19.04.2023)

8. HyperLedger Fabric. Fabric chaincode lifecycle. Access mode: https://hyperledger-fabric.readthedocs.io/en/latest/chaincode_lifecycle.html (Accessed 20.04.2023)

9. Ethereum Docs. Upgrading smart contracts. Access mode: https://ethereum.org/en/developers/docs/smart-contracts/upgrading/ (Accessed 20.04.2023)

10. Ethereum Improvement Proposals. ERC-1967: Proxy Storage Slots. Access mode: https://eips.ethereum.org/EIPS/eip-1967 (Accessed 20.04.2023)

11. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Patterns of object-oriented design / Translated from English: A. Slinkin. - SPb: Peter, 2021. - 448 p. - ISBN 978-5-4461-1595-2.

12. Ethereum Improvement Proposals. ERC-2535: Diamonds, Multi-Facet Proxy. Access mode: https://eips.ethereum.org/EIPS/eip-2535 (Accessed 20.04.2023).