ДЕЦЕНТРАЛИЗОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СЕРТИФИКАЦИИ СПЕЦИАЛИСТОВ НА ОСНОВЕ ТЕХНОЛОГИИ BLOCKCHAIN

В.А. Передерий, *студент*
М.Л. Рысин, *канд. пед. наук, доцент*
МИРЭА – Российский технологический университет
(Россия, г. Москва)

DOI:10.24412/2500-1000-2022-5-2-47-55

Аннотация. *В статье рассматриваются преимущества децентрализованной архитектуры и технологии Blockchain применительно к документоориентированным информационным системам профессиональных портфолио специалистов. Описаны методы проверки достоверности документов, целостности и конфиденциальности хранящихся в системе данных. Выводы проиллюстрированы на примере разработки прототипа такого рода информационной системы. В ходе реализации прототипа были проанализированы требования к системе, обоснованы архитектурные решения, представлен программный код одного из компонентов.*

Ключевые слова: *распределённые системы, децентрализованные системы, блокчейн, криптография, нереляционные базы данных, Use Case, UML.*

На рынке труда неизменно востребованными являются сертифицированные специалисты. Сертификация – это мероприятие по проверке уровня подготовки специалиста. По результатам сертификации в случае успеха специалисту выдаётся сертификат – документ, подтверждающий квалификацию в определённой области [1]. Сертификация чаще всего проводится по результатам прохождения различных образовательных программ.

Популярными становятся онлайн-платформы, которые предлагают курсы с получением цифрового сертификата, такие как GeekBrains, Skillbox, «Яндекс.Практикум» и другие.

Средством подтверждения выдачи сертификатов государственного образца является государственный реестр, где можно проверить отдельный документ, указав его данные. Однако не все компании имеют возможность зарегистрироваться как образовательное учреждение, а выдаваемые сертификаты не всегда имеют достаточную значимость, чтобы регистрировать их на государственном уровне в качестве диплома о дополнительном профессиональном образовании. Примером может послужить сертификат о победе в различного рода конкурсах (например, хакатонах).

В таком случае подтверждением достоверности занимается чаще всего само выдавшее сертификат юридическое лицо. В некоторых случаях для подтверждения документ подписывается цифровой подписью.

Это создаёт сложности как для рекрутеров, которые не имеют общего простого способа проверки подлинности сертификатов, так и для специалистов, которые вынуждены самостоятельно вести учёт полученных ими цифровых сертификатов и составлять своё портфолио. При отсутствии регистрации в реестрах также возникают сложности при аннулировании сертификатов. При этом утеря файла с сертификатом иногда может означать полную потерю сертификата, если выдавшая его сторона не хранит копию.

Существуют зарубежные сервисы, направленные на решение этих проблем, – это информационные системы, в задачи которых входит не только подтверждение достоверности хранящихся в них документов, но и их учёт. Одним из крупнейших является Credly Acclaim. Через данную систему многие зарубежные IT-компании выпускают электронные «значки», подтверждающие достижения обучающихся при прохождении образовательных программ в электронных образовательных

средах. В допандемийном 2019 году компания показала значительный рост, удвоив количество своих клиентов [2].

Системой Credly Acclaim пользуются такие зарубежные корпорации, как Cisco, Autodesk, Oracle, IBM, Amazon, Adobe и многие другие. Всё это доказывает востребованность подобного рода систем.

Авторам не известно о существовании подобных систем отечественного производства.

Централизованные и децентрализованные системы

Для реализации документо-ориентированной информационной системы подтверждения квалификаций специалистов необходимо сделать выбор между двумя принципиально разными подходами к построению её архитектуры. Система может быть централизованной либо децентрализованной.

Главным преимуществом *централизации* является производительность [3]. В зависимости от специфики системы, доступ к реестру сертификатов или средства для проверки достоверности может иметься, как только у партнёров (работодателей, учебных заведений и т.д.), так и у всех желающих. Централизованные системы обычно управляются одной компанией или лицом. Это обуславливает непрозрачность подобного рода систем и создаёт недоверие к ним.

Документы, которые можно хранить в централизованной системе, попадут в зависимость от лица, управляющего системой. Что, в свою очередь, открывает поле для злоупотреблений и коррупции.

В наше время распространение получили *децентрализованные* распределённые системы. Они отличаются от централизованных тем, что полная копия данных или их фрагмент может иметься у любого пользователя системы [4]. Данный подход увеличивает надёжность, так как копии или фрагменты данных имеются у большого количества пользователей, но вводит проблему сохранения конфиденциальности персональной информации. Ещё одна проблема заключается в том, что любой пользователь системы может исказить данные.

Для того чтобы решить эти проблемы, в распределённых системах применяются следующие технологии:

- Протоколы шифрования – позволяют избежать хранения данных в открытом виде, что позволяет сохранить конфиденциальность персональных данных пользователей.

- Технология Blockchain – позволяет избежать изменения данных, основываясь на свойствах хеш-функций и сложности задачи их расчёта.

Децентрализация негативно влияет на производительность системы, однако позволяет решить указанные выше проблемы централизованных систем.

Требования к системе

Рассмотрим бизнес-процесс подтверждения квалификации специалиста средствами децентрализованной Blockchain-системы (рис. 1). Подтверждение квалификации проводится путём получения специалистом соответствующего документа, подтверждающего квалификацию, добавления документа в информационную систему с последующей проверкой достоверности документа работодателем.

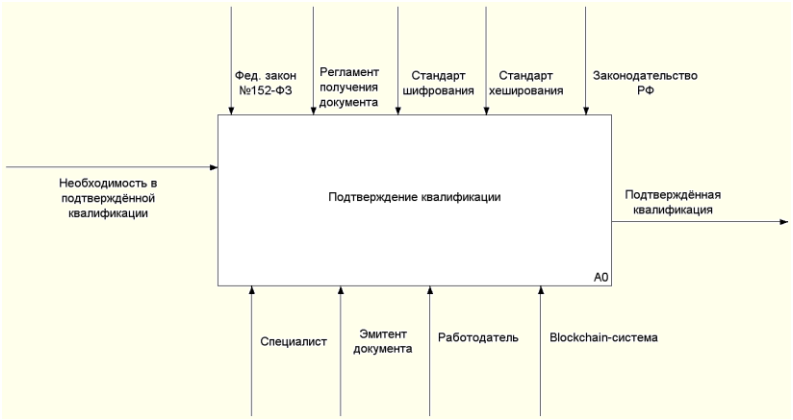

Рис. 1. Контекстная диаграмма процесса подтверждения квалификации специалиста средствами децентрализованной Blockchain-системы

Декомпозиция процесса «Подтверждение квалификации» изображена на рис. 2.

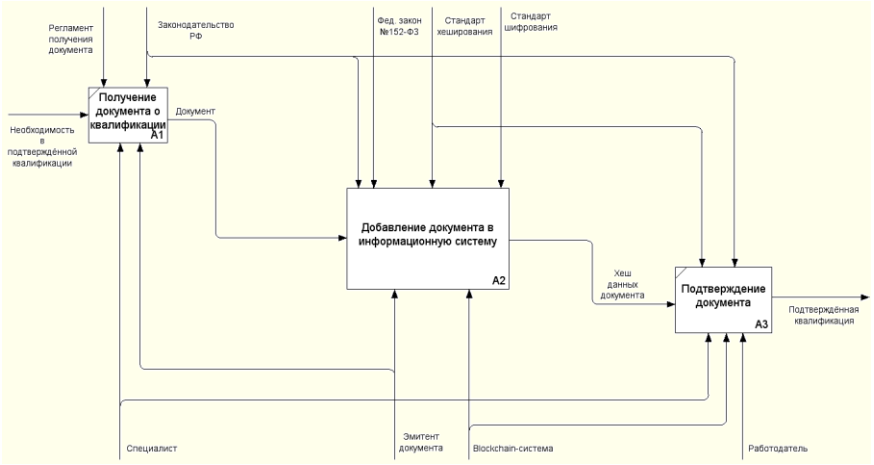

Рис. 2. Детализация процесса «Подтверждение квалификации»

Следующий уровень разбиения – декомпозиция процесса «Добавление документа в информационную систему» с учётом специфики децентрализованных систем изображён на рисунке 3.

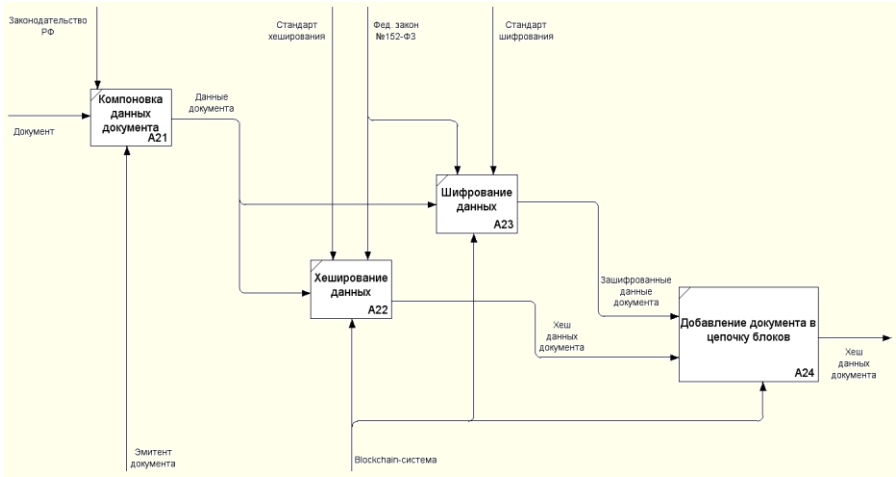

Рис. 3. Детализация процесса «Добавление документа в информационную систему»

В начале разработки любого программного средства требуется сформировать требования к нему, т.е. что пользователи системы ожидают от продукта, как с

функциональной, так и с других точек зрения [5].

Формирование требований является сложной задачей. Для упрощения процесса формирования требований существуют стандарты, одним из которых является язык моделирования Unified Modeling Language (далее – UML). Стандарт UML включает различные виды графических диаграмм, которые позволяют уточнить требования к информационной системе, а также перейти от общих словесных требований к особенностям реализации, начать составлять техническое задание для разработчиков [6].

Формирование требований к системе начнём с разработки UML-диаграммы Use Case. Актёры системы:

- Студент – специалист в определённой области, заинтересованный в повышении квалификации и формировании портфолио компетенций.
- Академия – организация, предоставляющая студентам образовательные услуги и выдающая сертификаты по окончании курсов.
- Рекрутёр – сотрудник компании, заинтересованный в найме квалифицированных кадров с подтверждёнными достижениями.
- Группа сопровождения – группа разработчиков системы, имеющая представителей от академий-партнёров, управляющая децентрализованной системой путём выпуска обновлений.

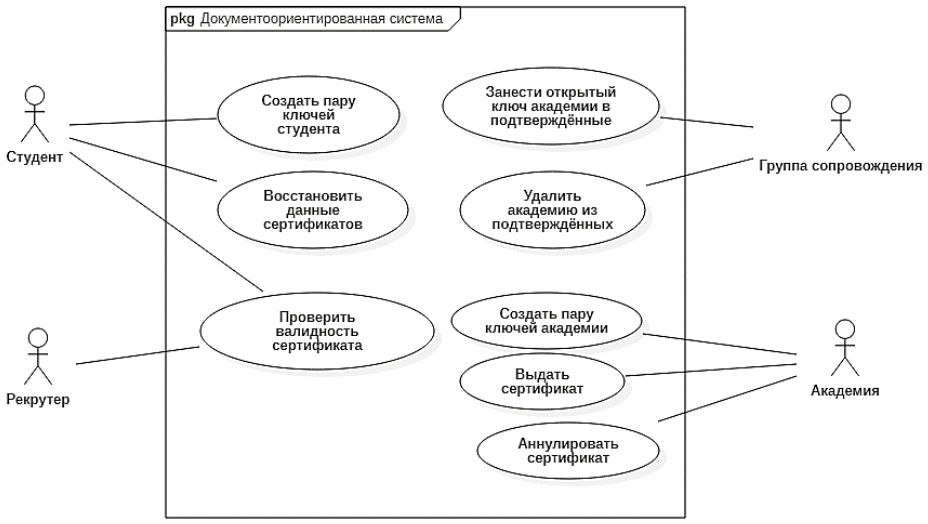

Рис. 4. Use Case диаграмма системы

Рассмотрим подробнее вариант использование «Восстановить данные сертификатов». Под сертификатом здесь подразумевается документ, подтверждающий компетенции. Система будет хранить данные сертификатов в виде блоков в цепочке. Блок в цепочке имеет следующий набор обязательных полей:

- открытый ключ (blockchain-адрес) академии (в открытом виде);
- открытый ключ (blockchain-адрес) получателя (в открытом виде);
- данные документа (в зашифрованном виде);
- хеш-сумма документа (в открытом виде);
- цифровая подпись академии для зашифрованных данных документа (в открытом виде);
- хеш-сумма предыдущего блока в цепочке (в открытом виде).

Как мы видим, в системе хранятся не личные данные пользователей, а лишь их открытые ключи. Документы, в свою очередь, шифруются. Таким образом, несмотря на то, что система является децентрализованной, конфиденциальность персональных данных пользователя сохраняется.

Когда пользователь передаёт работодателю свои документы, работодатель с помощью узла децентрализованной системы может хешировать документ и проверить наличие блока с такой хеш-суммой документа. Работодатель при этом может быть уверен, что все данные достоверны, благодаря технологии блокчейн. Работодатель так же может быть уверен, что сертификат действительно выдан указанной академией по наличию поля с цифровой подписью.

Специально для случая, когда пользователь хочет скачать свои данные из системы, в блоке хранятся данные документа, зашифрованные открытым ключом пользователя. Таким образом, пользователь может расшифровать данные своих сертификатов, используя свой закрытый ключ.

Произведём дальнейшее уточнение требований к системе, сформировав диаграмму деятельности варианта использования «Выдать сертификат» (рис. 5).

Архитектурные решения

Разрабатываемая информационная система должна иметь распределённую архитектуру. Такие системы включают в себя множество экземпляров разных или одинаковых приложений, в совокупности реализующих определённый сервис [4].

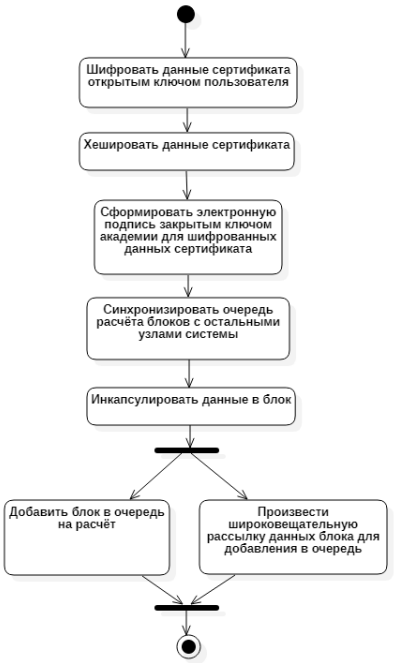

Рис. 5. Диаграмма деятельности варианта использования «Выдать сертификат»

Кроме этого, система должна быть децентрализованной. В таких системах каждый экземпляр приложения одновременно выступает как в роли клиента, так и в роли сервера. Фактически это ставит одну проблему – каждый пользователь для взаимодействия с системой будет также вынужден выполнять на своей рабочей станции сложную серверную логику системы.

Решением является выбор гибридной архитектуры, в которой узлами децентрализованной распределённой сети являются сервера, расположенные в организациях-операторах системы. А внутри организации-оператора пользователи взаимодействуют с системой по модели клиент-серверной архитектуры.

Предполагается и создание публичного сервера, доступного студентам академий посредством сети Интернет. Клиентское устройство может быть персональным компьютером, смартфоном или любым другим устройством, работающим с актуальными версиями веб-браузеров.

В Blockchain у каждого блока есть своя уникальная хеш-сумма, по которой осуществляется доступ к блоку [3]. Таким образом, для нашего приложения наиболее подходящей будет база данных вида «ключ-значение».

Базы данных вида «ключ-значение» является видом нереляционных баз данных, в которых доступ к данным производится по уникальным ключам. В нашем случае ключом будет строка с хеш-суммой блока, а значением – сериализованные данные блока. Кроме этого, нам потребуется служебное поле, в котором будет храниться хеш-сумма последнего блока в локальной копии цепочки блоков, а также длина цепочки. Ключами для данных полей будут строки «last_hash» и «blockchain_length» соответственно.

Так как база данных будет использоваться в сети только одним приложением, мы можем выбрать вариант локальной базы данных.

Пользовательский интерфейс системы

Особенности дизайна графического пользовательского интерфейса прототипа нашей системы отражены на рисунках 6-8.

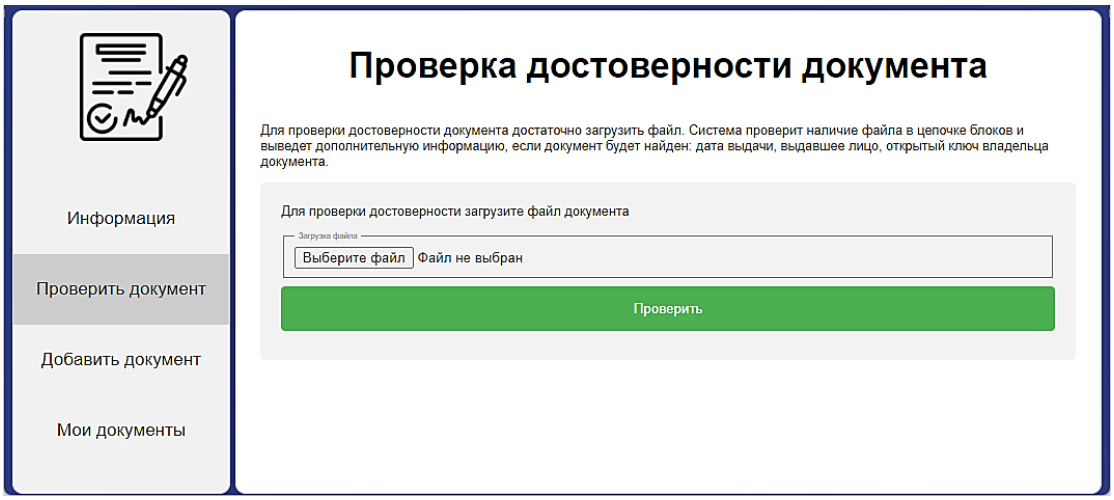

Рис. 6. Страница проверки достоверности документа

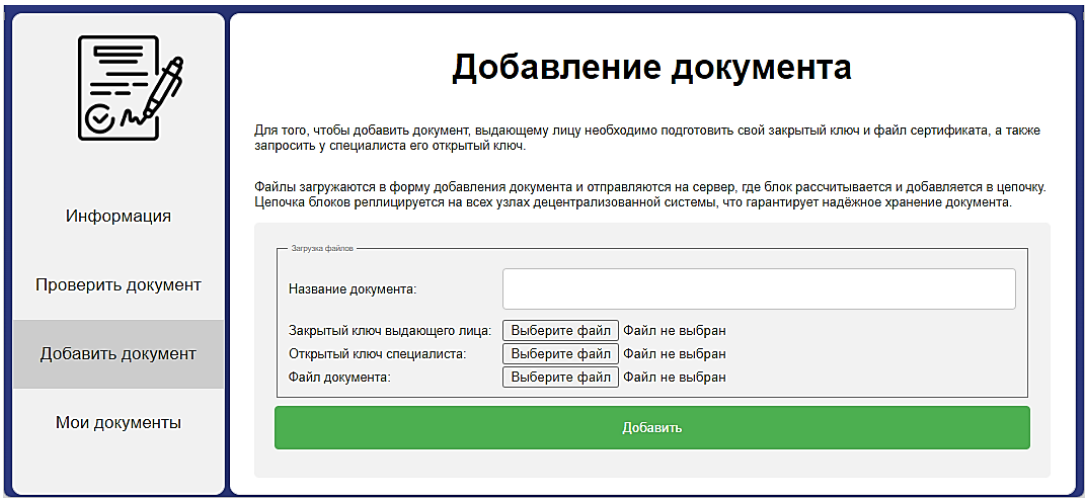

Рис. 7. Страница добавления документа в систему

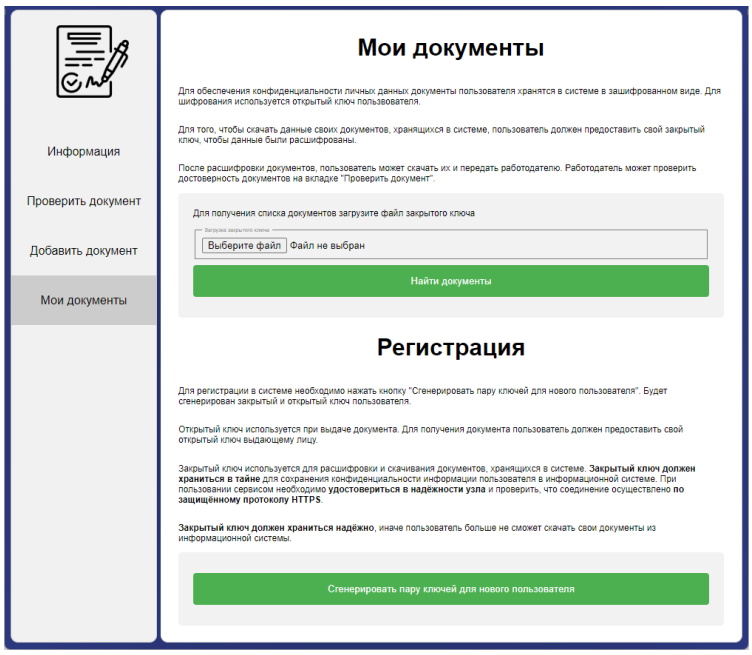

Рис. 8. Страница просмотра документов и регистрации

Некоторые практические аспекты реализации компонентов системы

Рассмотрим реализацию одного из обработчиков HTTP. Для регистрации пользователя сервер должен сгенерировать пару RSA-ключей и выслать их пользователю. Стоит отметить, что по HTTP можно за один запрос-ответ передать лишь один файл. При этом крайне нежелательно сохранять ключи в постоянную память, чтобы выслать их по-отдельности, так как в таком случае на диске могут оставаться следы файлов, даже после удаления.

Решением является создание двух файлов и объединение их в Zip-архив непосредственно в оперативной памяти. Это можно сделать с помощью библиотек «archive/zip», «bytes», «crypto/rsa», «crypto/rand», «crypto/x509», «encoding/pem», поставляемых вместе с языком Go.

Текст исходного кода обработчика на языке Go представлен в листинге 1.

В результате обработчик загружает архив keypair.zip, в котором находятся открытый и закрытый ключ пользователя в формате PEM.

Листинг 1. Исходный код функции.

```
func newKeypairHandler(w http.ResponseWriter, r *http.Request) {
    privateKey, err := rsa.GenerateKey(rand.Reader, 4096)
    if err != nil {
        w.WriteHeader(http.StatusInternalServerError)
        w.Write([]byte("Произошла ошибка при генерации пары ключей"))
        return
    }

    publicKey := privateKey.PublicKey

    var privateKeyBytes []byte = x509.MarshalPKCS1PrivateKey(privateKey)
    privateKeyBlock := &pem.Block{
        Type:  "RSA PRIVATE KEY",
        Bytes: privateKeyBytes,
    }
    privateKeyEncoded := pem.EncodeToMemory(privateKeyBlock)

    var publicKeyBytes []byte = x509.MarshalPKCS1PublicKey(&publicKey)
    publicKeyBlock := &pem.Block{
        Type:  "RSA PUBLIC KEY",
        Bytes: publicKeyBytes,
    }
    publicKeyEncoded := pem.EncodeToMemory(publicKeyBlock)

    buf := new(bytes.Buffer)
    zipWriter := zip.NewWriter(buf)
    var files = []struct {
        Name string
        Body []byte
    }{
        {"private.key", privateKeyEncoded},
        {"public.key", publicKeyEncoded},
    }

    for _, file := range files {
        zipFile, err := zipWriter.Create(file.Name)
        if err != nil {
            log.Println("Error when zipping keypair archive:", err)
            w.WriteHeader(http.StatusInternalServerError)
            w.Write([]byte("Произошла ошибка при генерации пары ключей"))
            return
        }
        _, err = zipFile.Write(file.Body)
        if err != nil {
            log.Println("Error when zipping keypair archive:", err)
            w.WriteHeader(http.StatusInternalServerError)
            w.Write([]byte("Произошла ошибка при генерации пары ключей"))
            return
        }
    }

    err = zipWriter.Close()
    if err != nil {
        log.Println("Error when zipping keypair archive:", err)
        w.WriteHeader(http.StatusInternalServerError)
        w.Write([]byte("Произошла ошибка при генерации пары ключей"))
        return
    }

    w.Header().Add("Content-Disposition", "Attachment; Filename=keypair.zip")
    http.ServeContent(w, r, "privateKey", time.Now(), bytes.NewReader(buf.Bytes()))
}
```

Заключение. Применение технологии Blockchain в информационных системах учёта документального подтверждения компетенций специалистов имеет значительные перспективы. Применение этой технологии позволит реализовать прозрачную и открытую информационную систему, на основе которой можно создать масштабный совместный проект с участием образовательных учреждений, центров сертификации и IT-компаний. При этом, в случае соблюдении ряда условий, система позволит гарантировать надёжность хранения и конфиденциальность занесённых в систему данных.

Библиографический список

1. Учебный центр при МГТУ им. Н.Э. Баумана Специалист.ru, «О центре тестирования и сертификации», 2022. – [Электронный ресурс]. – Режим доступа: https://www.specialist.ru/about-testingcenter (Дата обращения: 22.05.2022).

2. Credly Inc., «Общая информация о системе Credly Acclaim», 2021. – [Электронный ресурс]. – Режим доступа: https://info.credly.com/solution-for-product-certifications (Дата обращения: 22.05.2022).

3. Кравченко П. Блокчейн и децентрализованные системы: учебное пособие для студ. заведений высш. образования: в 3 частях. Ч. 1 / П. Кравченко, Б. Скрябин, О. Дубинина. – Харьков: ПРОМАРТ, 2018. – 400 с.

4. Бёрнс Б. Распределенные системы. Паттерны проектирования. – СПб.: Питер, 2019. – 224 с.

5. Рудаков А.В. Технология разработки программных продуктов. Практикум: учебное пособие / А.В. Рудаков, Г.Н. Федорова. – М.: Издательский центр «Академия», 2014. – 192 с.

6. Забродин А.В. Основы проектирования информационных систем с помощью языка UML: учебное пособие / А.В. Забродин, В.П. Бубнов. – СПб.: ПГУПС, 2018. – 46 с.

DECENTRALIZED INFORMATION SYSTEM FOR SPECIALISTS CERTIFICATION BASED ON BLOCKCHAIN TECHNOLOGY

V.A. Peredery, *Graduate Student*
M.L. Rysin, *Candidate of Pedagogical Sciences, Associate Professor*
MIREA – Russian Technological University
(Russia, Moscow)

Abstract. *The article discusses the advantages of decentralized architecture and Blockchain technology in relation to document-oriented information systems of specialists professional portfolio. Methods of verifying the reliability of documents, the integrity and configuration of the data stored in the system are described. The conclusions are illustrated by the example of that information system prototype development. When implementing the prototype, the requirements for the system were analyzed, architectural solutions were justified, the program code of one of the components was presented.*

Keywords: *distributed systems, decentralized systems, blockchain, cryptography, non-relating databases, Use Case, UML.*