

Politechnika Warszawska

W Y D Z I A Ł E L E K T R Y C Z N Y



Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych

Praca dyplomowa inżynierska

na kierunku Informatyka stosowana
w specjalności Inżynieria Oprogramowania

Wykorzystanie technologii blockchain do podpisywania autentyczności prac dyplomowych

Wojciech Szade

numer albumu 391110

promotor

dr inż. Robert Szmurło

WARSZAWA 2025

Wykorzystanie technologii blockchain do podpisywania autentyczności prac dyplomowych

Streszczenie

Niniejsza praca inżynierska dotyczy problemu weryfikacji autentyczności prac dyplomowych oraz możliwości jego rozwiązania z wykorzystaniem technologii blockchain. Na podstawie analizy istniejących metod uwierzytelniania dokumentów w środowisku akademickim zaproponowano autorskie, zdecentralizowane rozwiązanie oparte na modelu współpracy uczelni.

Praca zawiera przegląd wybranych technologii blockchain wykorzystywanych w systemach dystrybucji i potwierdzania danych, ze szczególnym uwzględnieniem ich przydatności do reprezentowania i podpisywania dokumentów akademickich. Szczegółowo opisano projekt i implementację funkcjonalnego prototypu systemu, obejmującego łańcuch bloków oparty na konsensusie Proof of Authority (PoA), warstwę komunikacji między węzłami, interfejs webowy umożliwiający dostęp do danych oraz aplikację desktopową służącą do niezależnej weryfikacji informacji.

System umożliwia bezpieczne i trwałe przechowywanie metadanych prac dyplomowych w rozproszonej strukturze blockchain, zapewniając ich integralność oraz dostępność dla różnych grup interesariuszy – zarówno w warunkach pełnego wsparcia uczelni, jak i po jego wycofaniu. Zastosowanie podpisów cyfrowych i tokenów JSON Web Token (JWT) pozwala na jednoznaczną identyfikację źródła danych oraz ich niezależne potwierdzenie bez konieczności odwoływania się do centralnego serwera.

Opracowane rozwiązanie ma charakter demonstracyjny i może stanowić podstawę do dalszego rozwoju rzeczywistego systemu weryfikacji dyplomów na poziomie międzyuczelnianym.

Słowa kluczowe: blockchain, weryfikacja autentyczności, Proof of Authority, system rozproszony

Using blockchain technology for authenticating diploma theses

Abstract

This engineering thesis addresses the problem of verifying the authenticity of academic theses and explores the potential of blockchain technology as a foundation for a decentralized verification system. Based on an analysis of existing document authentication methods in academic environments, an original blockchain-based solution was proposed, designed to operate within a network of cooperating universities.

The thesis presents an overview of selected blockchain technologies used in data distribution and verification systems, focusing on their applicability to representing and signing academic documents. It also describes the design and implementation of a functional prototype, including a custom blockchain with a Proof of Authority (PoA) consensus mechanism, peer-to-peer communication layer, web-based user interface, and a desktop application for independent data verification.

The proposed system enables secure and persistent storage of thesis metadata within a distributed ledger, ensuring data integrity and availability for various stakeholders – both while university support is maintained and after it is withdrawn. The use of digital signatures and JSON Web Tokens (JWTs) allows for reliable identification of data origin and offline validation without the need to contact a central server.

The developed solution serves as a proof of concept and may provide a foundation for further development of a real-world inter-university thesis verification system.

Keywords: blockchain, authenticity verification, Proof of Authority, distributed system

Spis treści

1	Wstęp	11
1.1	Problemy istniejących rozwiązań	11
1.2	Cel pracy i plan działania	12
2	Analiza istniejących rozwiązań	13
2.1	Rozwiązania wdrożone na polskich uczelniach	13
2.1.1	Szkoła Główna Handlowa – kwalifikowana pieczęć elektroniczna	13
2.1.2	Politechnika Świętokrzyska – weryfikacja dyplomów przy użyciu blockchain	14
2.1.3	Perspektywy i integracja z infrastrukturą zaufania	14
2.2	Systemy oparte na technologii blockchain	15
3	Potrzeby systemu	17
3.1	Aktorzy systemu	17
3.2	Potrzeby użytkowników	17
3.2.1	Student	17
3.2.2	Pracodawca	18
3.2.3	Pracownik uczelni	18
3.2.4	Administrator	19
3.3	Wymagania właściciela systemu	20
4	Proponowane rozwiązanie	21
4.1	Założenia projektowe	21
4.2	Bezpieczeństwo i mechanizm konsensusu	22
4.3	Token weryfikacyjny pracy dyplomowej	22
4.4	Zarządzanie danymi osobowymi	23
4.5	Podsumowanie	23
5	Architektura systemu	25
5.1	Moduły systemu	25
5.2	Struktura bloku	28
5.2.1	Pola przechowywane w bloku	29
5.2.2	Diagram klasy bloku	30

5.2.3	Opis metod klasy	30
5.2.4	Blok genezy	31
5.3	Procesy komunikacyjne w systemie	31
5.3.1	Uruchamianie systemu	31
5.3.2	Dodawanie nowego bloku	33
5.4	Protokoły komunikacji	35
5.4.1	Dodawanie nowych węzłów	35
5.4.2	Protokoły przesyłania wiadomości	36
6	Technologie	39
6.1	Język programowania	39
6.2	Baza danych	39
6.3	Backend aplikacji i API	39
6.4	Komunikacja Peer to Peer (P2P)	40
6.5	Operacje kryptograficzne	40
6.6	Interfejs użytkownika	40
6.6.1	Strona weryfikacyjna	40
6.6.2	Aplikacja do weryfikacji tokena	41
7	Instrukcja korzystania z systemu	43
7.1	Weryfikacja pracy dyplomowej	43
7.1.1	Weryfikacja za pomocą aplikacji desktopowej	44
7.2	Dodawanie nowej pracy dyplomowej	45
7.3	Instrukcja obsługi systemu dla administratorów	45
7.3.1	Konfiguracja wstępna	45
7.3.2	Zarządzanie węzłami	47
7.3.3	Synchronizacja łańcucha	47
8	Podsumowanie	49
8.1	Osiągnięte rezultaty	49
8.2	Napotkane ograniczenia i problemy	50
8.3	Możliwe usprawnienia i kierunki rozwoju	50
8.4	Wnioski końcowe	51
	Bibliografia	53
	Wykaz skrótów i symboli	55
	Słownik pojęć	57
	Spis rysunków	59
	Spis tabel	61

Rozdział 1

Wstęp

Fałszowanie dokumentów akademickich to poważny problem, który dotyka systemy edukacyjne na całym świecie [6]. Wzrost dostępności zaawansowanych technologii edycji i druku sprawia, że tworzenie fałszywych dyplomów i certyfikatów staje się coraz łatwiejsze. Ponadto, z powodu braku prostych metod weryfikacji, etap sprawdzania autentyczności dokumentów bywa często pomijany. W odpowiedzi na to rosnące zagrożenie, instytucje edukacyjne i pracodawcy poszukują nowoczesnych rozwiązań, które zapewnią wiarygodne potwierdzenie autentyczności dokumentów w dostępny dla użytkowników sposób.

Większość polskich uczelni udostępnia tzw. repozytoria prac dyplomowych, w których można wyszukać tytuły lub streszczenia obronionych prac. Rozwiązania te działają jednak wyłącznie w obrębie danej uczelni i nie tworzą jednolitego, ogólnokrajowego systemu weryfikacji. Jedynie nieliczne instytucje wprowadziły bardziej zaawansowane mechanizmy elektronicznej weryfikacji dyplomów – takie jak Szkoła Główna Handlowa w Warszawie, która stosuje elektroniczne pieczęcie do uwierzytelniania dyplomów [19], czy Politechnika Świętokrzyska, która wdrożyła rozwiązanie oparte na technologii blockchain [18]. Alternatywnie, niektóre uczelnie, jak Uniwersytet Gdański [4], stosują rozwiązania manualne, polegające na ręcznym potwierdzaniu autentyczności dokumentów w odpowiedzi na zapytania mailowe.

Na poziomie europejskim rozwijane są inicjatywy mające na celu ułatwienie cyfrowej weryfikacji kwalifikacji. Jednym z przykładów jest Europass Digital Credentials Infrastructure (EDCI), projekt umożliwiający wydawanie, przeglądanie oraz automatyczną weryfikację cyfrowych poświadczeń edukacyjnych w sposób interoperacyjny pomiędzy państwami członkowskimi Unii Europejskiej [3].

1.1 Problemy istniejących rozwiązań

W momencie pisania tej pracy inżynierskiej w Polsce nie funkcjonował żaden wspólny, zintegrowany system elektroniczny umożliwiający międzyuczelnianą weryfikację autentyczności dyplomów wydawanych absolwentom po obronie prac dyplomowych. Brak standaryzacji i jednolitej polityki utrzymania istniejących rozwiązań powoduje, że ich dostępność może być ograniczona lub ulec

zanikowi w przyszłości. Utrudnia to również dostęp pracodawcom, którzy – chcąc korzystać z takich rozwiązań – zmuszeni są do obsługi odrębnych procedur dla każdej uczelni.

W tym kontekście technologia blockchain zyskuje coraz większe zainteresowanie jako potencjalne rozwiązanie. Dzięki decentralizacji – eliminującej potrzebę centralnego organu kontrolnego – blockchain zapewnia unikalne cechy: większe bezpieczeństwo, transparentność oraz długowieczność systemu [5]. Niezmienność danych zapisanych w blockchainie gwarantuje ich integralność i autentyczność.

1.2 Cel pracy i plan działania

Badania pokazują, że blockchain ma potencjał aby zrewolucjonizować sposób, w jaki dokumenty akademickie są wydawane i weryfikowane [14]. Wdrożenie systemu opartego na blockchainie może przynieść wiele korzyści: zwiększenie zaufania do dokumentów, uproszczenie procesu weryfikacji, redukcję kosztów administracyjnych oraz stworzenie globalnie dostępnej platformy weryfikacyjnej.

Takie rozwiązanie musiałoby jednak sprostać kilku istotnym wyzwaniom:

- wysokiemu zużyciu energii w przypadku tradycyjnych algorytmów konsensusu [7],
- ryzyku przejęcia kontroli nad łańcuchem przez podmioty nieuprawnione [13], na przykład poprzez ataki typu 51% – które polegają na pozyskaniu kontroli nad większością węzłów systemu,
- zapewnieniu, aby dane udostępniane przez niezaufane źródła mogły być jednoznacznie weryfikowane względem zaufanego źródła prawdy.

Rozwiązanie tych problemów wymaga zastosowania dodatkowych mechanizmów zapewniających wiarygodność i integralność prezentowanych danych.

W dalszej części pracy autor szczegółowo przedstawi potrzeby i ograniczenia systemu umożliwiającego weryfikację autentyczności prac dyplomowych. Czytelnikowi zostanie przybliżone pojęcie blockchaina oraz jego podstawowe komponenty. Następnie przeprowadzona zostanie analiza zalet i wad zastosowania technologii blockchain w kontekście omawianego problemu. Autor porówna również istniejące mechanizmy konsensusu i wskaże te, które najlepiej odpowiadają wymaganiom projektu. Omówione zostaną także przykłady już istniejących rozwiązań wykorzystujących blockchain w podobnych zastosowaniach.

Po zrozumieniu i przedstawieniu tych zagadnień autor zaproponuje konkretne technologie i rozwiązania, które zostaną wykorzystane do stworzenia prototypowego systemu. Implementacja zostanie przeprowadzona w częściowo uproszczonej formie, ze szczególnym naciskiem na przedstawienie architektury oraz sposobu działania zaprojektowanego systemu.

System powstały w wyniku tej pracy będzie mógł stanowić drogowskaz dla uczelni wyższych, które – współpracując ze sobą – mogłyby stworzyć oparty na zaprezentowanym modelu, zunifikowany system weryfikacji prac dyplomowych.

Rozdział 2

Analiza istniejących rozwiązań

Przed zaprojektowaniem własnego rozwiązania konieczne jest przeanalizowanie istniejących systemów, które realizują podobne zadania lub posługują się zbliżonymi technologiami. Pozwoli to lepiej zrozumieć dostępne możliwości, a także ograniczenia, jakie mogą się pojawić podczas projektowania i wdrażania systemu.

W pierwszej kolejności omówione zostaną rozwiązania stosowane na wybranych polskich uczelniach wyższych. Następnie przedstawiona zostanie analiza systemów opartych na technologii blockchain, które mogą stanowić punkt odniesienia przy budowie własnej architektury.

Dokładne przyjrzenie się istniejącym rozwiązaniom umożliwi określenie, które mechanizmy warto wykorzystać, a które wymagają innego podejścia, dostosowanego do specyfiki systemu opisywanego w tej pracy.

2.1 Rozwiązania wdrożone na polskich uczelniach

2.1.1 Szkoła Główna Handlowa – kwalifikowana pieczęć elektroniczna

Szkoła Główna Handlowa w Warszawie (SGH) wdrożyła rozwiązanie umożliwiające wydawanie absolwentom cyfrowych kopii dyplomów opatrzonych kwalifikowaną pieczęcią elektroniczną [16]. Pieczęć elektroniczna funkcjonuje podobnie jak podpis elektroniczny, jednak jest przypisana do instytucji, a nie do osoby fizycznej. Uczelnia podpisuje dokumenty kwalifikowanym certyfikatem, co umożliwia ich późniejszą weryfikację z użyciem ogólnodostępnych narzędzi. Rozwiązanie to jest zgodne z europejskimi standardami identyfikacji elektronicznej eIDAS i opiera się na zaufanej infrastrukturze Public Key Infrastructure (PKI).

Absolwent otrzymuje od SGH plik PDF zawierający dyplom lub suplement podpisany elektronicznie. Weryfikacja autentyczności dokumentu jest możliwa na przykład za pomocą programu Adobe Reader, który sprawdza integralność podpisu i potwierdza tożsamość wystawcy dokumentu na podstawie europejskich list zaufania.

Rozwiązanie to charakteryzuje się kilkoma istotnymi zaletami. Weryfikacja dokumentu jest prosta, niewymagająca specjalistycznego oprogramowania ani zaawansowanej wiedzy od użytkownika

końcowego. System bazuje na uznanej infrastrukturze zaufania, co ułatwia akceptację dokumentów zarówno w Polsce, jak i w innych krajach Unii Europejskiej. Wadą tego podejścia jest jednak zależność od funkcjonowania list zaufanych dostawców oraz konieczność zarządzania ważnością certyfikatów.

W porównaniu do proponowanego w pracy rozwiązania opartego o blockchain, system SGH opiera się na centralnym modelu zaufania, podczas gdy blockchain eliminuje potrzebę zaufanej trzeciej strony. System bazujący na blockchainie umożliwia niezależną weryfikację autentyczności dokumentu na podstawie danych przechowywanych w rozproszonym rejestrze. Zaletą rozwiązania SGH jest natychmiastowa zgodność z obowiązującymi regulacjami i prostota wdrożenia. Wdrożenie technologii blockchain wymagałoby z kolei odpowiedniego uwzględnienia wymogów prawnych, ale oferowałoby większą niezależność i odporność na awarie pojedynczych instytucji.

2.1.2 Politechnika Świętokrzyska – weryfikacja dyplomów przy użyciu blockchain

Politechnika Świętokrzyska (PŚ) wdrożyła rozwiązanie umożliwiające zapis cyfrowych kopii dyplomów w sieci blockchain przy współpracy z firmą Billon [11]. Zamiast podpisywać dokumenty kwalifikowaną pieczęcią, uczelnia zapisuje ich zawartość w zdecentralizowanym rejestrze danych. Absolwent otrzymuje od uczelni unikalny identyfikator lub link oraz klucz dostępu do odczytu swojego dyplomu.

Technologia wykorzystana w tym rozwiązaniu zapisuje pełną treść dokumentu w blockchainie. Dane są rozpraszane pomiędzy różne węzły sieci, co zabezpiecza je przed nieautoryzowaną modyfikacją lub usunięciem. Dostęp do odczytu dokumentu jest kontrolowany za pomocą klucza prywatnego.

Weryfikacja dokumentu odbywa się przez wprowadzenie danych na dedykowanej stronie internetowej udostępnionej przez uczelnię. System umożliwia szybkie i bezpieczne potwierdzenie autentyczności dokumentu w sposób zdalny.

Zaletami tego rozwiązania są wysoka odporność na fałszerstwa oraz decentralizacja danych. Brak konieczności pośrednictwa instytucji certyfikujących eliminuje dodatkowe kroki w procesie uwierzytelniania dokumentu. Pewnym ograniczeniem może być konieczność wykorzystania dedykowanego narzędzia udostępnionego przez uczelnię oraz potrzeba zarządzania dostępem do kluczy.

W porównaniu do rozwiązania proponowanego w pracy, wdrożenie PŚ korzysta z komercyjnej technologii zewnętrznego dostawcy. Proponowany system miałby charakter otwarty i umożliwiałby wspólne zarządzanie przez wiele uczelni, co zwiększałoby jego niezależność. W autorskim rozwiązaniu przewidziano również alternatywne podejście do przechowywania danych: zamiast pełnych dokumentów zapisywane byłyby ich skróty kryptograficzne, co zmniejszałoby wymagania dotyczące przestrzeni i zwiększało skalowalność systemu.

2.1.3 Perspektywy i integracja z infrastrukturą zaufania

Analiza powyższych systemów wskazuje, że choć autorskie rozwiązania (jak dedykowana aplikacja weryfikacyjna blockchain PŚ czy prototyp z niniejszej pracy) zapewniają nowy poziom innowacyjności, to istniejące ramy prawne i infrastrukturalne również oferują sprawdzone podejścia do weryfikacji

dyplomów. W szczególności wykorzystanie kwalifikowanych pieczęci elektronicznych i europejskich list zaufanych dostawców usług (Trusted Lists UE) stanowi alternatywę, która już dziś gwarantuje uznawalność dokumentów w skali międzynarodowej. Przykład SGH pokazuje, że dyplom podpisany pieczęcią kwalifikowaną może zostać zweryfikowany przez każdego odbiorcę przy użyciu ogólnodostępnych narzędzi, bez potrzeby odrębnej aplikacji – zaufanie wynika tu z certyfikatu wystawionego przez podmiot znajdujący się na publicznej liście zaufania [16]. Można zatem rozważyć, czy rozwijając systemy blockchain do weryfikacji dokumentów, nie warto łączyć ich z tą infrastrukturą. Jednym z możliwych kierunków integracji jest opatrywanie kluczowych elementów rozwiązania (np. wpisów o wydaniu dyplomu w blockchain) kwalifikowanym podpisem lub pieczęcią uczelni. Dzięki temu informacje w rozproszonym rejestrze zyskałyby formalny status zaufany – byłyby sygnowane przez instytucję edukacyjną w sposób weryfikowalny przy użyciu list zaufania. Innym aspektem jest wykorzystanie list zaufanych dostawców do automatycznej weryfikacji tożsamości uczelni w systemie: zamiast polegać wyłącznie na identyfikatorach w blockchain, aplikacja weryfikacyjna mogłaby sprawdzać, czy certyfikat użyty do podpisania bloku lub transakcji pochodzi od kwalifikowanego dostawcy (co zwiększałoby wiarygodność takiej platformy w oczach użytkowników).

Podsumowując, kwalifikowane pieczęcie elektroniczne wsparte unijnym systemem zaufania stanowią silną podstawę prawną i technologiczną dla uwierzytelniania dokumentów, z której już korzystają uczelnie (jak SGH). Rozwiązanie proponowane w pracy wychodzi jednak naprzeciw potrzebie dalszej cyfryzacji i decentralizacji – eliminuje niektóre ograniczenia (np. potrzebę centralnego poświadczania każdego dokumentu osobno) i dodaje nowe możliwości (jak natychmiastowa weryfikacja wielu dyplomów w jednym rejestrze czy odporność na pojedynczy punkt awarii dzięki decentralizacji). Włączenie do tego systemu elementów z ekosystemu zaufania (pieczęci kwalifikowanych, list zaufania) może w przyszłości ułatwić jego akceptację i integrację z istniejącymi procesami, łącząc zalety obu podejść – tradycyjnego (opartego na PKI) i nowatorskiego (opartego na blockchain).

2.2 Systemy oparte na technologii blockchain

Technologia blockchain jest wszechstronna i może być dostosowywana do różnorodnych zastosowań. Systemy oparte na tej technologii, nawet jeśli służą podobnym celom, mogą znacznie różnić się pod względem przyjętych rozwiązań oraz wynikających z nich zalet i ograniczeń.

Jednym z najbardziej znanych zastosowań technologii blockchain jest bitcoin – kryptowaluta oparta na rozproszonej księdze rachunkowej, umożliwiająca realizację płatności bez udziału centralnych instytucji finansowych [15]. Bitcoin wykorzystuje mechanizm konsensusu *Proof-of-Work* i został zaprojektowany jako zdecentralizowany system płatności, a nie jako uniwersalna platforma dla aplikacji [8]. Zasada jego działania opiera się na zapisywaniu transakcji w blokach, tworzących trwałą, niezmienną rejestr.

Z uwagi na zastosowany mechanizm konsensusu, sieć Bitcoin charakteryzuje się wysokim zużyciem energii elektrycznej [7], co stanowi przedmiot krytyki. Nowoczesne platformy blockchainowe, takie

jak *Ethereum*, przechodzą na bardziej energooszczędne algorytmy konsensusu, aby zmniejszyć wpływ technologii na środowisko.

Systemem bardziej zbliżonym funkcjonalnie do rozważanego w pracy rozwiązania jest Bigchain DB w wersji 2.0.

BigchainDB 2.0 łączy cechy typowe dla blockchainu, takie jak decentralizacja i niezmiennosc danych, z właściwościami baz danych: wysoką szybkością operacji, niskimi opóźnieniami oraz możliwością indeksowania i wyszukiwania informacji [1].

Głównym założeniem BigchainDB 2.0 było połączenie zalet obu technologii przy jednoczesnym ograniczeniu ich wad. We wcześniejszych wersjach pojawiały się problemy związane ze scentralizowanym zarządzaniem oraz podatnością na awarie pojedynczych węzłów. Wersja 2.0 wprowadziła decentralizację i zwiększoną odporność systemu [1].

Do kluczowych cech BigchainDB 2.0 należą:

- **Pełna decentralizacja:** każdy węzeł posiada własną bazę danych *MongoDB*, a komunikacja w sieci realizowana jest przy pomocy protokołu Tendermint. System zachowuje funkcjonalność nawet przy awarii części węzłów [1].
- **Niezmiennosc danych:** dane zapisane w sieci są trwałe i niepodlegające modyfikacjom, a próby ich zmiany są łatwe do wykrycia [1].
- **Wysoka wydajność:** system umożliwia realizację dużej liczby transakcji na sekundę, przy jednoczesnym niskim opóźnieniu ich zatwierdzenia [1].
- **Wyszukiwanie i indeksowanie danych:** dzięki wykorzystaniu możliwości *MongoDB*, dane mogą być efektywnie przeszukiwane i filtrowane [1].
- **Odporność na ataki Sybil:** dostęp do sieci mają jedynie autoryzowane węzły, co zwiększa bezpieczeństwo systemu [1].

BigchainDB 2.0 znajduje zastosowanie w takich dziedzinach jak zarządzanie łańcuchami dostaw, ochrona praw własności intelektualnej czy przechowywanie i zarządzanie danymi [1]. Łącząc właściwości blockchainu i baz danych, system ten oferuje elastyczne możliwości budowy rozproszonych, bezpiecznych aplikacji.

BigchainDB 2.0 wykorzystuje algorytm konsensusu Raft, który opiera się na wyborze lidera koordynującego zapisy transakcji. W przypadku awarii lidera pozostałe węzły przeprowadzają głosowanie w celu wyboru nowego lidera [10].

Mimo wielu zalet, BigchainDB 2.0 nie spełnia wszystkich wymagań stawianych przed systemem projektowanym w ramach niniejszej pracy. W szczególności, system nie umożliwia zarządzania uprawnieniami węzłów, co stanowi kluczową funkcjonalność dla modelu opartego na zaufanych uczelniach wyższych [1]. Dodatkowym ograniczeniem jest złożoność systemu oraz trudność jego pełnej integracji z istniejącymi rozwiązaniami uczelnianymi.

Pomimo tych różnic, BigchainDB 2.0, dzięki otwartemu kodowi źródłowemu i rozbudowanej dokumentacji, stanowi cenne źródło wiedzy oraz inspiracji przy projektowaniu własnego rozwiązania.

Rozdział 3

Potrzeby systemu

Po przeanalizowaniu problemu opisanego w rozdziale 1 należy określić wymagania dla systemu opartego o technologię blockchain, który mógłby rozwiązać zidentyfikowane problemy. Pierwszym krokiem jest ustalenie grup użytkowników systemu – aktorów – oraz zrozumienie ich potrzeb względem projektowanego rozwiązania.

3.1 Aktorzy systemu

W systemie wyróżniono następujących aktorów:

- **Student** – autor pracy dyplomowej, którego praca została dodana do systemu. Jego celem jest przedstawienie dowodu autentyczności swojej pracy potencjalnemu odbiorcy.
- **Pracodawca** – odbiorca dowodu autentyczności pracy, na przykład rekruter lub przedstawiciel innej uczelni wyższej.
- **Pracownik uczelni** – osoba odpowiedzialna za wprowadzanie danych o pracach dyplomowych do systemu oraz przekazywanie studentowi informacji niezbędnych do weryfikacji pracy.
- **Administrator** – osoba zarządzająca funkcjonowaniem systemu oraz węzłami sieci blockchain.

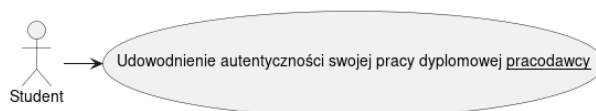
Dodatkowo stroną odpowiedzialną za decyzję o wdrożeniu i utrzymaniu systemu jest **właściciel systemu** (uczelnia wyższa), pełniący rolę interesariusza.

3.2 Potrzeby użytkowników

3.2.1 Student

Celem studenta jest udowodnienie autentyczności swojej pracy dyplomowej pracodawcy lub innemu podmiotowi. W ramach systemu student otrzyma od uczelni kryptograficzny odcisk dokumentu, czyli unikalny skrót kryptograficzny (*hash*) wyliczony na podstawie treści pracy. Odcisk ten będzie służył jako dowód autentyczności dokumentu. W dalszej części pracy pojęcia *kryptograficzny odcisk dokumentu* oraz *hash pracy* będą używane zamiennie.

Student będzie przekazywał pracodawcy odcisk dokumentu, a także – opcjonalnie – treść swojej pracy w formacie PDF, umożliwiając pełną weryfikację dokumentu. Diagram przypadków użycia studenta przedstawiono na rysunku 1.

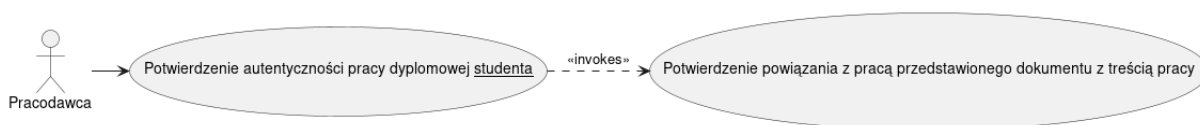


Rysunek 1. Diagram przypadków użycia studenta

3.2.2 Pracodawca

Podstawową funkcjonalnością dla pracodawcy będzie możliwość sprawdzenia autentyczności pracy dyplomowej. System powinien zapewniać szybki dostęp do niezbędnych informacji oraz generować precyzyjne i jednoznaczne komunikaty. Istotne jest, aby system minimalizował ryzyko błędnej interpretacji wyników weryfikacji.

W przypadku udostępnienia także pełnej treści pracy w formacie PDF, pracodawca będzie mógł zweryfikować zgodność przesłanego dokumentu z zapisanym w systemie odciskiem. Diagram przypadków użycia pracodawcy przedstawiono na rysunku 2.



Rysunek 2. Diagram przypadków użycia pracodawcy

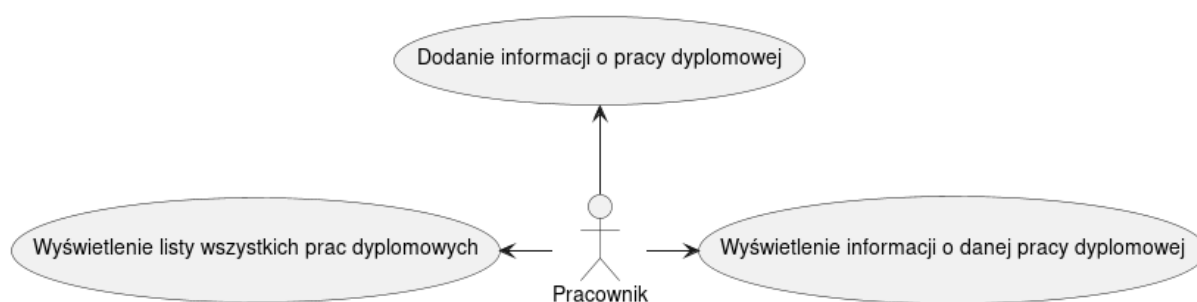
3.2.3 Pracownik uczelni

Pracownik uczelni będzie odpowiedzialny za ręczne dodawanie danych o nowych pracach dyplomowych do systemu. W idealnym przypadku proces ten mógłby być zautomatyzowany, jednak na potrzeby projektu zakłada się ręczne wprowadzanie danych. Pracownik uczelni powinien mieć możliwość:

- dodawania informacji o nowej pracy dyplomowej,
- przeglądania listy wprowadzonych prac,
- wyświetlania szczegółowych informacji o konkretnych pracach.

System powinien weryfikować, czy dodawana praca nie jest już obecna w rejestrze oraz informować o niepowodzeniach przy dodawaniu danych. Ze względu na charakterystykę propagacji bloków w technologii blockchain, czas dodania pracy może być zmienny i zależy od wybranego mechanizmu konsensusu. Przyjęto, że w typowych warunkach dodanie pracy nie powinno trwać dłużej niż 24 godziny.

Diagram przypadków użycia pracownika uczelni przedstawiono na rysunku 3.



Rysunek 3. Diagram przypadków użycia pracownika uczelni

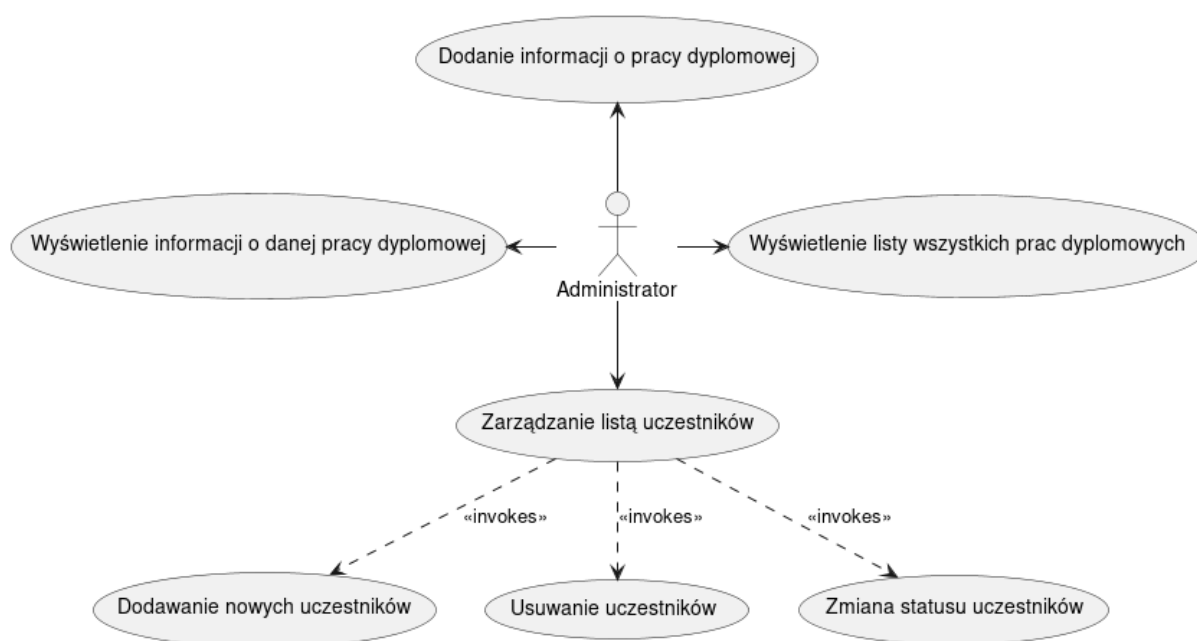
3.2.4 Administrator

Administrator systemu będzie nadzorował jego działanie. Administrator powinien mieć możliwość:

- przeglądania informacji o pracach dyplomowych,
- zarządzania listą węzłów (dodawanie, usuwanie, zmiana statusu),
- monitorowania integralności sieci i reagowania na nieprawidłowości.

Administrator nie będzie miał możliwości edycji ani usuwania już wprowadzonych danych, co jest zgodne z fundamentalnymi założeniami technologii blockchain.

Diagram przypadków użycia administratora przedstawiono na rysunku 4.



Rysunek 4. Diagram przypadków użycia administratora

3.3 Wymagania właściciela systemu

Właściciel systemu (uczelnia) oczekuje od rozwiązania spełnienia kluczowych wymagań:

- wysokiego poziomu bezpieczeństwa,
- możliwości integracji z istniejącymi systemami uczelni,
- umiarkowanych wymagań sprzętowych i energetycznych,
- odporności na próby fałszowania danych oraz utraty dostępności systemu.

Rozdział 4

Proponowane rozwiązanie

W odpowiedzi na zidentyfikowane potrzeby systemu (rozdział 3) oraz analizę istniejących rozwiązań (rozdział 2), autor zdecydował się na opracowanie własnego rozwiązania, które spełnia specyficzne wymagania dotyczące weryfikacji autentyczności prac dyplomowych z wykorzystaniem technologii blockchain.

System składa się z trzech głównych komponentów:

- **Backend systemu** – zarządzający łańcuchem bloków, weryfikacją danych i obsługą zapytań.
- **Strona internetowa** – udostępniająca prosty interfejs użytkownika umożliwiający weryfikację autentyczności pracy na podstawie identyfikatora pracy (hasha).
- **Aplikacja weryfikacyjna** – pozwalająca na lokalną weryfikację tokena weryfikacyjnego pracy dyplomowej na podstawie podpisu cyfrowego uczelni.

Takie rozdzielenie systemu umożliwia zarówno łatwą integrację z platformami uczelni w czasie aktywnego wsparcia projektu, jak i niezależną weryfikację prac po zakończeniu oficjalnego funkcjonowania systemu.

4.1 Założenia projektowe

Projekt opiera się na następujących założeniach funkcjonalnych:

- W systemie uczestniczy wiele uczelni wyższych, które dodają prace dyplomowe do wspólnego łańcucha blockchain.
- Po dodaniu pracy do łańcucha, nie ma możliwości jej edycji ani usunięcia.
- Autor pracy otrzymuje od uczelni unikalny identyfikator (hash) swojej pracy, który może przekazać potencjalnym odbiorcom.

Przyjęte w projekcie założenia techniczne:

- System powinien zapewniać długowieczność przechowywanych danych, nawet w przypadku zaprzestania wsparcia ze strony uczelni, poprzez możliwość utrzymywania kopii łańcucha przez innych uczestników sieci.

- Opracowana wersja systemu stanowi *Proof of Concept*, mający na celu demonstrację koncepcji, z możliwością dalszego rozwoju w pełnoprawne rozwiązanie.
- Interfejs użytkownika służy głównie do prezentacji działania systemu; docelowo system powinien być zintegrowany z istniejącymi rozwiązaniami uczelni.

4.2 Bezpieczeństwo i mechanizm konsensusu

W celu zapewnienia bezpieczeństwa i integralności danych zastosowano mechanizm konsensusu PoA. W tym modelu tylko zaufane węzły, reprezentujące uczelnie wyższe, mogą tworzyć nowe bloki w łańcuchu. Lista kluczy publicznych uprawnionych węzłów jest zapisana w bloku genezy (*ang. genesis block*) – bloku, który jest pierwszym elementem łańcucha i zawiera inną strukturę niż pozostałe bloki. Dzięki zawarciu w nim kluczy, każdy uczestnik sieci może zweryfikować autentyczność nowo dodanego bloku.

Dodanie bloku przebiega według schematu:

1. Uczelnia generuje blok zawierający dane pracy dyplomowej.
2. Blok jest podpisywany prywatnym kluczem uczelni.
3. Blok jest rozgłaszany do pozostałych węzłów w sieci.
4. Węzły weryfikują podpis za pomocą kluczy publicznych, a następnie w przypadku poprawności, dodają blok do swojego łańcucha.

Zastosowanie PoA umożliwia zachowanie bezpieczeństwa bez konieczności stosowania energochłonnych lub czasochłonnych algorytmów konsensusu, takich jak *Proof of Work*.

4.3 Token weryfikacyjny pracy dyplomowej

Aby umożliwić niezależną od uczelni weryfikację autentyczności pracy, w systemie stosowany jest **token weryfikacyjny pracy dyplomowej** bazujący na standardzie JWT.

Token ten zawiera pełną zawartość bloku, w tym informacje takie jak imię i nazwisko autora, tytuł pracy, nazwę uczelni, numer indeksu, a także dane techniczne bloku, obejmujące jego własny hash oraz hash poprzedniego bloku. Całość jest podpisywana kluczem prywatnym uczelni, która wystawiła dyplom.

Weryfikacja tokena przebiega w kilku krokach:

1. Pracodawca otrzymuje od studenta token weryfikacyjny.
2. Aplikacja weryfikująca odczytuje nazwę uczelni z nagłówka tokena i na tej podstawie wybiera odpowiedni klucz publiczny z bloku genezy.
3. Aplikacja weryfikuje poprawność podpisu cyfrowego tokena.

4. Po pomyślnej weryfikacji wyświetlane są wszystkie dane zawarte w bloku.

Takie rozwiązanie umożliwia potwierdzenie autentyczności pracy bez konieczności łączenia się z serwerami uczelni. Wymaga ono jedynie udostępniania przez dowolną z uczelni możliwości pobrania aplikacji ze strony w domenie uczelni lub jej oficjalnego githuba.

Warto zauważyć, że proponowane podejście wykorzystujące aplikację lokalną mogłoby być również zastąpione lub rozbudowane o bardziej scentralizowane rozwiązanie. Przykładowo, podobnie jak w systemach pieczęci elektronicznych stosowanych w instytucjach Unii Europejskiej, możliwe byłoby utrzymywanie centralnej bazy kluczy publicznych lub uruchomienie dedykowanej usługi weryfikacyjnej na poziomie ogólnokrajowym lub międzynarodowym. Takie podejście umożliwiłoby potwierdzanie autentyczności prac bez potrzeby instalowania aplikacji lokalnej.

Obecna implementacja została jednak świadomie zaprojektowana jako rozwiązanie zdecentralizowane, które nie wymaga istnienia pojedynczego zaufanego organu, a dzięki temu pozostaje odporne na ewentualne wycofanie wsparcia przez poszczególne instytucje.

4.4 Zarządzanie danymi osobowymi

W obecnej wersji projektu dane osobowe autorów (imię, nazwisko, numer indeksu) przechowywane są w blockchainie w jawnej postaci. Jest to konieczne, aby zapewnić możliwość jednoznacznej identyfikacji autorów prac i zapobiec podszywaniu się pod inne osoby.

W przyszłości w bardziej dojrzałych rozwiązaniach można rozważyć:

- zastąpienie jawnych danych ich skrótami kryptograficznymi,
- zastosowanie mechanizmów typu *zero-knowledge proofs* dla zwiększenia prywatności,
- przechowywanie danych poza blockchainem i odwoływanie się do nich poprzez referencje.

4.5 Podsumowanie

Proponowane rozwiązanie łączy:

- bezpieczeństwo i niezawodność technologii blockchain,
- model konsensusu oparty na zaufaniu do uczelni (PoA),
- wykorzystanie JWT do umożliwienia weryfikacji autentyczności pracy bez konieczności dostępu do centralnego serwera.

Projekt został opracowany jako baza do dalszego rozwoju i integracji z systemami informatycznymi uczelni wyższych.

Rozdział 5

Architektura systemu

W tym rozdziale przedstawiona została architektura systemu blockchainowego zaprojektowanego do weryfikacji autentyczności prac dyplomowych. Omówiono podział systemu na moduły, sposób ich wzajemnej komunikacji, a także fizyczną strukturę rozmieszczenia komponentów w środowisku uruchomieniowym.

5.1 Moduły systemu

Na rysunku 5 przedstawiono ogólną architekturę systemu z podziałem na główne komponenty oraz zależności pomiędzy nimi. Komponenty zostały podzielone na dwa pakiety: **Frontend** oraz **Backend**.

Pakiet **Frontend** obejmuje komponent *Web app*, który udostępnia interfejs użytkownika systemu. Z interfejsu tego korzystać będą studenci, pracodawcy, pracownicy uczelni oraz administratorzy. Komunikacja *Web app* z zapleczem systemu odbywa się za pośrednictwem *API*.

Pakiet **Backend** składa się z kilku głównych komponentów odpowiedzialnych za logikę systemu, zarządzanie blockchainem i komunikację w sieci:

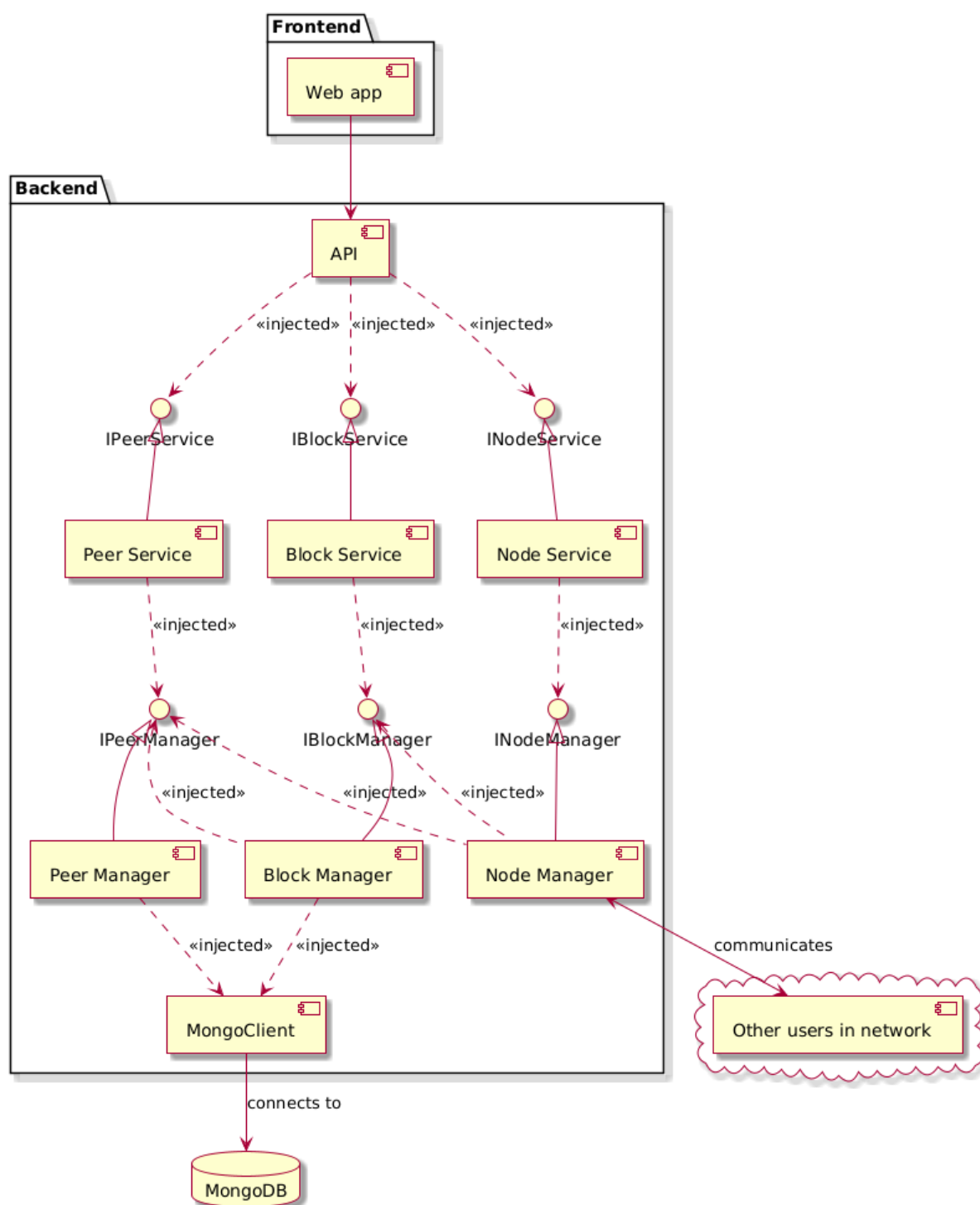
- **API** – komponent obsługujący komunikację z interfejsem użytkownika (*Frontend*). Umożliwia odczyt danych z blockchajna, dodawanie nowych bloków oraz zarządzanie połączeniami z innymi węzłami. API korzysta z trzech głównych serwisów: *IPeerService*, *INodeService* i *IBlockService*.
- **Peer Service** – warstwa pośrednicząca pomiędzy API a *Peer Managerem*. Odpowiada za zarządzanie listą znanych węzłów, w tym dodawanie nowych uczestników sieci, ich blokowanie i usuwanie oraz modyfikowanie ich statusów.
- **Node Service** – pośrednik pomiędzy API a *Node Managerem*. Obsługuje operacje wymagające komunikacji z innymi węzłami, takie jak synchronizacja łańcucha, generowanie nowych bloków oraz inicjalizacja połączeń sieciowych.
- **Block Service** – warstwa obsługująca operacje na lokalnym łańcuchu bloków: odczyt danych, generowanie bloku genezy oraz usuwanie istniejących bloków (np. w celu ponownej synchronizacji).

- **Peer Manager** – komponent odpowiedzialny za przechowywanie i zarządzanie informacjami o znanych węzłach. Przechowuje adresy, klucze publiczne, statusy i pozwala na dynamiczną aktualizację stanu sieci.
- **Node Manager** – moduł realizujący komunikację typu *Peer-to-Peer* między węzłami. Obsługuje synchronizację łańcucha, propagację nowych bloków oraz inicjalizację połączeń z innymi uczestnikami sieci.
- **Block Manager** – komponent zarządzający strukturą blockchaina w ramach lokalnej bazy danych. Tworzy nowe bloki, weryfikuje ich poprawność oraz umożliwia manipulowanie łańcuchem.
- **MongoDB** – baza danych dokumentowa przechowująca wszystkie bloki łańcucha oraz informacje o znanych węzłach sieci.

Dodatkowo, poza główną strukturą systemu, opracowano niezależną aplikację służącą do lokalnej weryfikacji autentyczności prac dyplomowych. Aplikacja ta, zbudowana przy użyciu biblioteki *Tkinter*, umożliwia użytkownikom sprawdzanie poprawności podpisu tokenu weryfikacyjnego pracy dyplomowej (JWT) bez konieczności łączenia się z systemem. Ze względu na swoją pomocniczą funkcję, aplikacja nie została ujęta w diagramach architektury.

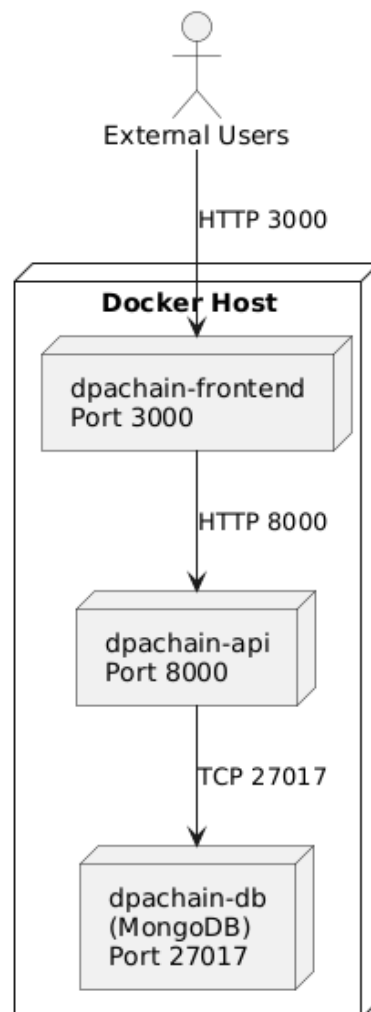
Diagram komunikacji pomiędzy powyższymi komponentami przedstawiono na rysunku 5.

Dodatkowo na rysunku 6 przedstawiono architekturę fizyczną systemu – sposób rozmieszczenia głównych komponentów w środowisku uruchomieniowym.



Rysunek 5. Diagram komponentów architektury systemu

Physical Architecture Diagram



Rysunek 6. Diagram architektury fizycznej systemu

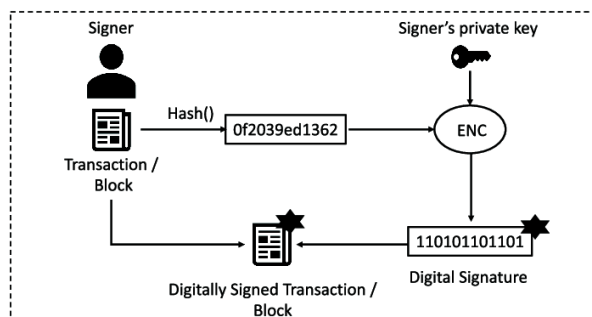
5.2 Struktura bloku

Informacje o pracach dyplomowych w systemie przechowywane są w postaci bloków. Każda praca dyplomowa jest reprezentowana przez jeden blok w łańcuchu, a bloki uporządkowane są w kolejności dodawania. Wszystkie przynależą do wspólnego łańcucha (*blockchain*).

Każdy blok posiada własny identyfikator (*hash*), wyliczany na podstawie zawartości bloku oraz hasha poprzedniego bloku. Kluczową cechą tej struktury jest niezmiennosc danych: zmiana jakiegokolwiek informacji w bloku powodowałaby zmianę jego hasha, co z kolei unieważniałoby łańcuch.

Dodatkowo, każdy blok zawiera podpis cyfrowy, będący podpisanym haszem bloku. Podpis ten generowany jest za pomocą klucza prywatnego węzła tworzącego blok, a następnie może być

zweryfikowany przez inne węzły za pomocą publicznego klucza danego uczestnika [12]. Proces podpisywania bloku przedstawiono na rysunku 7.



Rysunek 7. Proces podpisywania bloku [12]

Struktura bloku została zaprojektowana w sposób umożliwiający jego późniejszą rozbudowę, dlatego każdy blok zawiera również informację o wersji łańcucha.

Blok przechowuje dodatkowo **token weryfikacyjny pracy dyplomowej** w formacie JWT, który zawiera zakodowaną pełną zawartość bloku. Token ten jest podpisywany przez uczelnię i służy do późniejszej niezależnej weryfikacji autentyczności danych.

5.2.1 Pola przechowywane w bloku

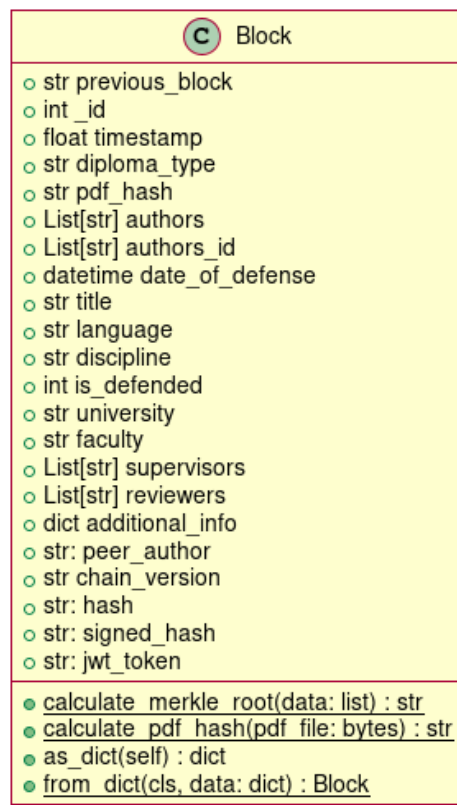
Blok zawiera następujące pola:

- **previous_block** – hash poprzedniego bloku,
- **_id** – identyfikator bloku (kolejny numer),
- **timestamp** – znacznik czasu utworzenia bloku,
- **diploma_type** – rodzaj dyplomu (np. licencjacki, magisterski),
- **pdf_hash** – hash pliku PDF pracy dyplomowej,
- **authors** – lista autorów pracy,
- **authors_id** – lista numerów identyfikacyjnych autorów (np. numer indeksu),
- **date_of_defense** – data obrony pracy,
- **title** – tytuł pracy,
- **language** – język pracy,
- **discipline** – dziedzina nauki,
- **is_defended** – status obrony pracy (tak/nie),
- **university** – nazwa uczelni,
- **faculty** – wydział uczelni,
- **supervisors** – promotorzy pracy,
- **reviewers** – recenzenci pracy,
- **additional_info** – dodatkowe informacje,
- **peer_author** – węzeł tworzący blok,

- **chain_version** – wersja łańcucha,
- **hash** – hash bloku,
- **signed_hash** – podpisany hash bloku,
- **jwt_token** – token weryfikacyjny pracy dyplomowej (JWT).

5.2.2 Diagram klasy bloku

Diagram klasy Block, przedstawiający strukturę bloku oraz najważniejsze metody operacyjne, zamieszczono na rysunku 8.



Rysunek 8. Diagram klasy bloku

5.2.3 Opis metod klasy

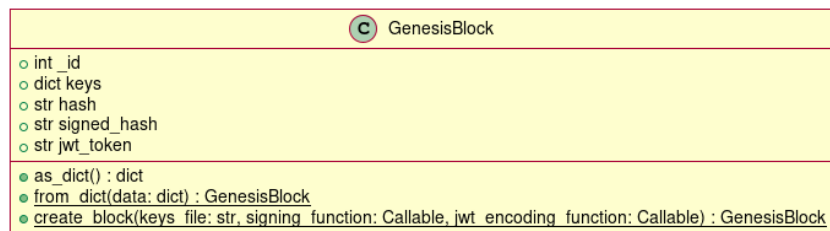
Klasa bloku udostępnia kilka kluczowych metod:

- `calculate_merkle_root(data: list)` – oblicza korzeń drzewa Merkle na podstawie listy danych. Zapewnia integralność zawartości bloku.
- `calculate_pdf_hash(pdf_file: bytes)` – generuje hash z pliku PDF pracy dyplomowej.
- `as_dict(self)` – zwraca reprezentację bloku w postaci słownika (*dictionary*).
- `from_dict(cls, data: dict)` – tworzy instancję obiektu na podstawie przekazanego słownika.

5.2.4 Blok genezy

Pierwszy blok w łańcuchu, tzw. blok genezy, różni się istotnie od pozostałych bloków. Nie przechowuje on informacji o pracy dyplomowej, lecz zawiera dane konfiguracyjne umożliwiające uruchomienie i walidację łańcucha – w szczególności publiczne klucze autoryzowanych węzłów. Dzięki temu każdy nowy węzeł może samodzielnie, bez dodatkowych źródeł zaufania, zweryfikować autoryzację innych uczestników sieci. W założeniu – blok ten jest dystrybuowany jako część każdego systemu.

Blok genezy zawiera również podpisany hash swoich danych oraz token JWT, który może być używany do niezależnej walidacji zgodności bloku w środowiskach offline. Bloku genezy nie zawiera odwołań do poprzedniego bloku i zawsze ma numer `_id = 0`. Struktura tej klasy została przedstawiona na rysunku 9.



Rysunek 9. Diagram klasy bloku genezy

5.3 Procesy komunikacyjne w systemie

5.3.1 Uruchamianie systemu

Na rysunku 10 przedstawiono diagram sekwencji procesu uruchamiania systemu. Ilustruje on przepływ komunikacji pomiędzy komponentami od momentu startu aplikacji.

Pierwszą akcją, wykonywaną przez administratora bezpośrednio na maszynie serwera, jest uruchomienie systemu. Ponieważ pierwotnie system jeszcze nie działa, komunikacja poprzez panel administratora ani API nie jest jeszcze możliwa.

Po inicjalizacji system nawiązuje połączenie z bazą danych MongoDB. Połączenie to umożliwia zarówno odczyt, jak i modyfikację danych.

Następnym krokiem jest uruchomienie API dla panelu administratora. Od tego momentu możliwe staje się wysyłanie zapytań administracyjnych, takich jak synchronizacja łańcucha.

Kolejnym etapem jest uruchomienie instancji węzła sieci P2P, który otwiera port na połączenia przychodzące od innych uczestników sieci.

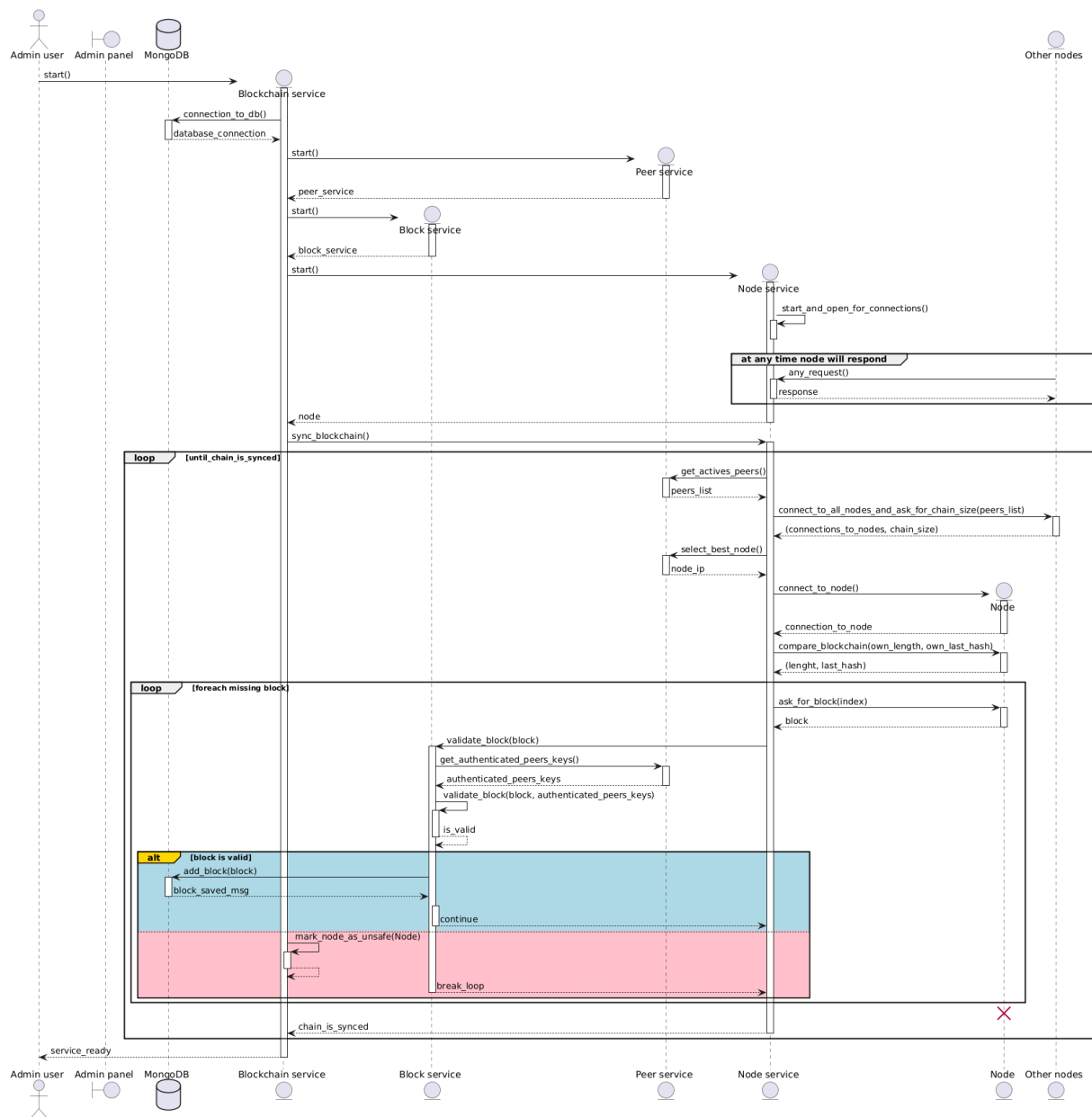
Po aktywacji węzła następuje proces synchronizacji lokalnego łańcucha z innymi węzłami:

- Węzeł nawiązuje połączenia z innymi znanymi węzłami i pobiera od nich informacje o długości ich łańcuchów.

- Na podstawie długości łańcucha oraz wiarygodności węzła (autoryzacji) wybierany jest najlepszy węzeł do synchronizacji. Wybór jest dodatkowo częściowo losowy, aby uniknąć przeciążenia jednego uczestnika sieci.
- Węzeł pobiera brakujące bloki, weryfikuje ich podpisy i spójność z dotychczasowym łańcuchem. W przypadku wykrycia nieprawidłowości, synchronizacja z danym węzłem jest przerywana, a węzeł oznaczany jest jako niewiarygodny.

Po zakończeniu synchronizacji system wysyła do pozostałych uczestników sieci (którzy mogą mieć niepełny łańcuch) polecenie synchronizacji. Węzły te samodzielnie decydują, czy wykonają synchronizację, weryfikując wiarygodność nadawcy i analizując aktualny stan swojego łańcucha.

Po zakończeniu wszystkich etapów administrator otrzymuje informację o pełnej gotowości systemu do pracy.



Rysunek 10. Diagram sekwencji procesu uruchamiania aplikacji

5.3.2 Dodawanie nowego bloku

Diagram przedstawiony na rysunku 11 ilustruje proces dodawania nowego bloku reprezentującego pracę dyplomową do łańcucha.

W tym scenariuszu system, API oraz panel administratora są już aktywne. Administrator lub pracownik uczelni, posiadający odpowiednie uprawnienia, wprowadza nowe dane pracy dyplomowej poprzez interfejs użytkownika.

Kolejne kroki procesu:

- API przekazuje żądanie dodania pracy do usługi zarządzającej blockchainem.

5.4 Protokoły komunikacji

Komunikacja pomiędzy węzłami sieci realizowana jest przy użyciu zaprojektowanych protokołów komunikacyjnych. Biblioteka *p2pd*, opisana w rozdziale 6, umożliwia przesyłanie danych pomiędzy węzłami w formacie binarnym.

Aby zapewnić poprawną obsługę przesyłanych informacji, opracowano spójny standard wiadomości oraz zestaw protokołów komunikacji.

Każda wiadomość przesyłana w systemie posiada zunifikowaną strukturę:

- **protokół** – określający typ wiadomości oraz sposób jej obsługi po stronie odbiorcy,
- **autor** – nazwę węzła wysyłającego wiadomość,
- **podpis** – podpis cyfrowy wiadomości, utworzony przy użyciu klucza prywatnego węzła,
- **ładunek** – opcjonalne dane przesyłane w treści wiadomości (np. blok).

Proces przygotowania wiadomości obejmuje:

1. Serializację danych do postaci JSON Canonicalization Scheme (JCS) – uporządkowanej formy JSONa umożliwiającego poprawne podpisywanie i weryfikację wiadomości.
2. Wygenerowanie podpisu poprzez zahaszowanie danych w formacie JCS i podpisanie ich przy użyciu klucza prywatnego nadawcy.
3. Dodanie prefiksu "DPACHAIN" – oznaczającego wiadomości wysłane przy użyciu naszego wewnętrznego protokołu komunikacji – i zakodowanie wiadomości do formatu binarnego.

Stosowana metoda podpisywania wiadomości jest zbliżona do idei JWT, ponieważ umożliwia odbiorcy niezależną weryfikację autentyczności nadawcy wiadomości na podstawie podpisu i znanego klucza publicznego, bez konieczności utrzymywania aktywnej sesji lub dodatkowych form uwierzytelnienia.

Każdy węzeł odpowiada na wiadomości przychodzące od znanych i niezablokowanych nadawców. Węzły nieznane lub uprzednio zidentyfikowane jako złośliwe mogą zostać zablokowane. Decyzje o blokadzie podejmuje administrator systemu na podstawie zarejestrowanej historii komunikacji oraz analizy zachowań węzłów. Możliwe jest w przyszłości zautomatyzowanie procesu blokowania węzłów, które np. wysyłają nadmierną liczbę wiadomości, próbują wprowadzać nieprawidłowe informacje do łańcucha lub dopuszczają się innych działań uznanych za złośliwe.

Po odebraniu wiadomości węzeł powinien przestać odpowiadać (jeśli protokół tego wymaga), a następnie zakończyć połączenie. Zdecydowano się na model krótkotrwałych połączeń, aby uniknąć utrzymywania niepotrzebnych wątków komunikacyjnych oraz zmniejszyć ryzyko przeciążenia węzła odpowiadającego.

5.4.1 Dodawanie nowych węzłów

System umożliwia przyłączanie nowych węzłów na dwa sposoby:

- **Ręczne dodawanie** – administrator systemu ręcznie wprowadza informacje o nowych węzłach. Wyłącznie tą metodą dodawane są węzły autoryzowane, posiadające uprawnienia do tworzenia nowych bloków.
- **Samodzielne zgłaszanie się** – węzeł nieautoryzowany może wysłać wiadomość przy użyciu specjalnego protokołu przedstawiającego się do znanego mu węzła, którego adres musi wcześniej pozyskać w sposób ręczny (np. poprzez publikację na stronie projektu).

W obecnej wersji projektu zgłoszenia samodzielne są automatycznie akceptowane. W przyszłości możliwe będzie wprowadzenie dodatkowych mechanizmów ochrony, takich jak ręczne zatwierdzanie nowych zgłoszeń, blokowanie zgłoszeń z określonych adresów IP lub ograniczenie możliwości dołączania wyłącznie do zatwierdzonych węzłów uczelnianych.

Lista znanych węzłów nie jest automatycznie rozpowszechniana pomiędzy uczestnikami sieci. Decyzję o rezygnacji z takiego mechanizmu podjęto ze względu na charakter sieci opartej na zaufanych i stosunkowo stałych węzłach uczelni. W praktyce brak automatycznego udostępniania listy nie wpływa negatywnie na działanie systemu, ponieważ wystarczy, aby każdy węzeł utrzymywał połączenia z kilkoma znanymi uczestnikami.

W ramach obecnie projektowanego systemu możliwe jest również odzyskanie informacji o części węzłów autoryzowanych bezpośrednio z danych przechowywanych w blockchainie. Każdy blok zawiera pole `peer_author`, w którym zapisywany jest identyfikator (adres sieciowy) węzła, który utworzył dany blok. Dzięki temu możliwe jest częściowe odtworzenie wiedzy o aktywnych uczestnikach, dopóki uczelnie utrzymują swoje węzły.

W sytuacji, w której uczelnie zrezygnowałyby z dalszego udziału w projekcie, brak mechanizmu propagacji listy węzłów mógłby stać się istotnym ograniczeniem. W takim scenariuszu należałoby rozważyć wprowadzenie automatycznej wymiany informacji o znanych węzłach w ramach komunikacji P2P, aby zwiększyć odporność sieci na zanikanie aktywnych uczestników.

Wyszukiwanie węzłów jest klasycznym wyzwaniem w sieciach rozproszonych. Typowe podejścia obejmują wykorzystywanie tzw. węzłów bootstrapowych – dobrze znanych i utrzymywanych punktów wejściowych do sieci – lub mechanizmy propagowania informacji o znanych węzłach między uczestnikami. Wprowadzenie podobnych rozwiązań w opisywanym projekcie byłoby naturalnym krokiem w kierunku zapewnienia stabilności sieci w warunkach ograniczonego udziału uczelni.

5.4.2 Protokoły przesyłania wiadomości

Protokoły przesyłania wiadomości w systemie można podzielić na dwie główne grupy:

- **Protokoły zapytań i odpowiedzi** – para wiadomości – zapytanie o dane i odpowiedź zawierająca wymagane informacje:
 - `ask_chain_size / response_chain_size` – zapytanie i odpowiedź dotycząca rozmiaru łańcucha bloków,
 - `ask_compare_blockchain / response_compare_blockchain` – zapytanie i odpowiedź zawierająca długość oraz hash ostatniego bloku w łańcuchu,

- ask_block / response_block – zapytanie o konkretny blok i jego przesłanie w odpowiedzi.
- **Protokoły jednostronne** – wiadomości nieoczekujące odpowiedzi:
 - new_peer – zgłoszenie nowego węzła do sieci,
 - ask_sync_chain – prośba o synchronizację łańcucha bloków.

Protokoły specjalne odgrywają kluczową rolę w rozwoju i utrzymaniu sieci: pozwalają na samodzielne zgłaszanie nowych uczestników oraz utrzymywanie aktualnej kopii łańcucha u wszystkich aktywnych węzłów.

Rozdział 6

Technologie

W tym rozdziale omówiono technologie i narzędzia wykorzystane przy budowie systemu blockchain do weryfikacji autentyczności prac dyplomowych. Wybór poszczególnych rozwiązań był podyktowany kilkoma kryteriami: stabilnością i dojrzałością technologii, dostępnością bibliotek oraz dokumentacji, łatwością integracji, a także możliwością dalszej rozbudowy projektu.

6.1 Język programowania

Głównym językiem programowania projektu został **Python**. Oferuje on wysoką czytelność kodu oraz dostęp do szerokiego ekosystemu bibliotek wspierających tworzenie aplikacji webowych, operacje kryptograficzne oraz podstawowe funkcjonalności blockchained. Python umożliwia szybkie prototypowanie rozwiązań bez konieczności ręcznego zarządzania zasobami, co odróżnia go od języków niższego poziomu.

6.2 Baza danych

Do przechowywania bloków wybrano nierelacyjną bazę danych **MongoDB**. Struktura dokumentowa w formacie *JSON* jest naturalnie dostosowana do sposobu zapisu danych w blockchainie, gdzie każdy blok przechowuje kompletną jednostkę informacji. MongoDB zapewnia wysoką wydajność, prostą skalowalność oraz elastyczność struktury danych [2]. Baza ta jest także wykorzystywana w rozwiązaniu BigchainDB 2.0 [1], opisanym w rozdziale 2.

6.3 Backend aplikacji i API

Do budowy backendu i udostępnienia API wybrano framework **FastAPI**. Początkowo rozważano użycie *Django*, jednak jego ścisłe powiązanie z relacyjnymi bazami danych sprawiłoby, że integracja z *MongoDB* byłaby utrudniona lub wymagała zastosowania dodatkowych, niewspieranych narzędzi, takich jak *Djongo* [9].

Alternatywą było wykorzystanie *Flaska*, jednak ostatecznie wybrano *FastAPI* ze względu na nowocześniejszą architekturę, automatyczne generowanie dokumentacji oraz wbudowaną walidację danych. Porównanie FastAPI i Flask przedstawiono w tabeli 1.

Kryterium	FastAPI	Flask
Wydajność	Wysoka – dzięki natywnemu wsparciu dla programowania asynchronicznego; idealna dla aplikacji o dużej liczbie równoczesnych połączeń.	Niższa – przez domyślną synchroniczność; wsparcie dla asynchroniczności wymaga dodatkowych bibliotek, takich jak <i>Gevent</i> .
Dokumentacja API	Automatycznie generowana (<i>Swagger UI</i> , <i>ReDoc</i>).	Wymaga dodatkowych narzędzi, np. <i>Flask-RESTX</i> .
Walidacja danych	Wbudowana walidacja (<i>Pydantic</i>).	Wymaga użycia dodatkowych bibliotek, takich jak <i>Marshmallow</i> .
Przypadki użycia	Optymalny dla mikrousług i wydajnych API.	Wszechstronny dla prostych aplikacji webowych.

Tabela 1. Porównanie bibliotek FastAPI i Flask [17]

6.4 Komunikacja P2P

Do komunikacji między węzłami systemu wybrano bibliotekę **p2pd**. W ekosystemie Pythona jest to obecnie jedyna szeroko dostępna biblioteka umożliwiająca stabilną i zaawansowaną obsługę połączeń *Peer-to-Peer*. Oferuje między innymi funkcje wyszukiwania węzłów, negocjacji połączeń oraz utrzymywania sesji komunikacyjnych.

6.5 Operacje kryptograficzne

Operacje kryptograficzne, takie jak podpisywanie bloków, generowanie hashy dokumentów oraz weryfikacja podpisów cyfrowych, realizowane są przy pomocy biblioteki **pycryptodomex**. Biblioteka ta jest uznawana za standard w środowisku Python dla zaawansowanych operacji kryptograficznych.

6.6 Interfejs użytkownika

6.6.1 Strona weryfikacyjna

Do stworzenia strony internetowej służącej do weryfikacji prac dyplomowych wykorzystano środowisko **Node.js** wraz z frameworkiem *Express*. Wybór ten podyktowany był:

- szybkością tworzenia prostych serwerów HTTP,
- łatwością integracji z technologiami frontendowymi,
- elastycznością rozwoju i potencjalnej integracji z systemami uczelnianymi.

6.6.2 Aplikacja do weryfikacji tokena

W celu umożliwienia lokalnej weryfikacji autentyczności pracy dyplomowej, opracowano aplikację przy użyciu **Tkinter** – standardowej biblioteki GUI dla Pythona. Tkinter umożliwia szybkie tworzenie prostych, lekkich aplikacji działających na systemach Windows, Linux oraz macOS bez potrzeby instalowania dodatkowych zależności. Zastosowanie tego samego języka programowania co w backendzie systemu pozwoliło na łatwe odwzorowanie wymaganej funkcjonalności oraz zachowanie spójności technologicznej projektu.

Rozdział 7

Instrukcja korzystania z systemu

W tym rozdziale przedstawiona zostanie instrukcja obsługi systemu, przeznaczona dla wszystkich jego użytkowników.

W pierwszej kolejności omówione zostaną przypadki użycia opisane w rozdziale 3, ze szczególnym uwzględnieniem interfejsu użytkownika, z którego korzystają pracodawcy i pracownicy uczelni. W dalszej części skupiono się na działaniach administratora systemu. Zgodnie z założeniami projektu, rolę administratora może przyjąć dowolny użytkownik – co jest szczególnie istotne w sytuacji, gdy uczelnie wycofają swoje wsparcie dla systemu. W takim przypadku konieczne jest posiadanie przez użytkownika podstawowych umiejętności technicznych oraz odpowiedniego środowiska sprzętowego. System został zaprojektowany z myślą o możliwie najprostszej konfiguracji i obsłudze, poprzez zastosowanie konteneryzacji (Docker), gotowych skryptów startowych oraz rozbudowanej walidacji danych, w tym danych konfiguracyjnych.

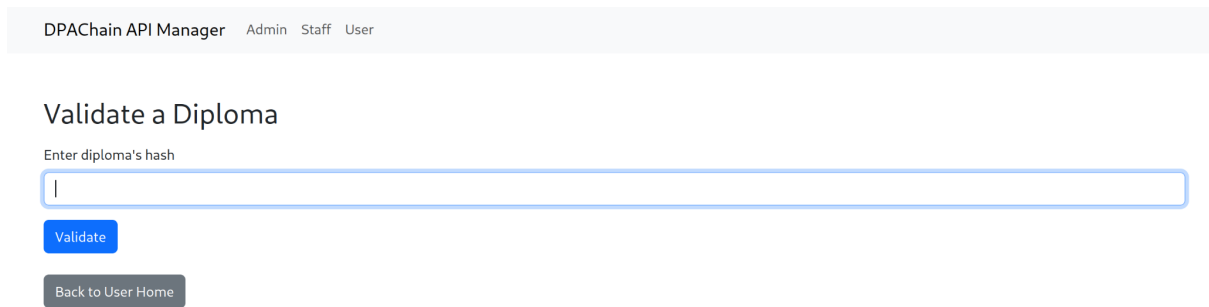
7.1 Weryfikacja pracy dyplomowej

Student przedstawiający swoje kompetencje pracodawcy udostępnia mu kryptograficzny odcisk dokumentu powiązany z jego pracą dyplomową. Opcjonalnie może również przekazać samą pracę w formacie PDF.

Pracodawca, posiadając te informacje, odwiedza stronę służącą do walidacji dyplomów. Jeśli system jest wciąż wspierany, strona ta będzie znajdowała się w domenie jednej z uczelni uczestniczących w systemie. W przypadku zaprzestania wsparcia, walidacja może odbywać się za pośrednictwem strony prowadzonej przez podmiot zewnętrzny.

Przykładowy wygląd strony przedstawiono na rysunku 12. Interfejs przygotowano w języku angielskim, jednak możliwe jest jego przetłumaczenie lub integracja z własnymi rozwiązaniami uczelni poprzez API.

Po wprowadzeniu hasha i kliknięciu `Validate`, system sprawdza istnienie odpowiadającego bloku w łańcuchu. Jeśli taki blok nie istnieje, użytkownik otrzyma komunikat jak na rysunku 13.



DPACChain API Manager Admin Staff User

Validate a Diploma

Enter diploma's hash

[Validate](#)

[Back to User Home](#)

Rysunek 12. Strona do weryfikacji dyplomu



DPACChain API Manager Admin Staff User

Validate a Diploma

Enter diploma's hash

[Validate](#)

A diploma with specified hash doesn't exists

[Back to User Home](#)

Rysunek 13. Komunikat o nieistnieniu żądanej pracy

W przypadku pozytywnej weryfikacji wyświetlane są szczegóły pracy oraz hash pliku PDF. Użytkownik może przesłać swój plik w celu porównania go z zapisanym hashem. Przykład pozytywnej walidacji przedstawiono na rysunku 14.

7.1.1 Weryfikacja za pomocą aplikacji desktopowej

Jeśli użytkownik nie ufa stronie internetowej lub chce zweryfikować token JWT samodzielnie, może skorzystać z aplikacji desktopowej. Aplikację można pobrać z oficjalnych repozytoriów uczelni. Udostępnione są wersje dla Windowsa, Linuxa i MacOS.

W aplikacji wprowadza się otrzymany token, a po kliknięciu przycisku **Authenticate** następuje jego weryfikacja. Po poprawnej weryfikacji wyświetlane są dane o pracy. Dodatkowo można sprawdzić zgodność pliku PDF. Interfejs aplikacji przedstawiono na rysunku 15.

Diploma Details
Valid ✓

Title: Wykorzystanie technologii blockchain do podpisywania autentyczności prac dyplomowych
Authors: Wojciech Szade (319110)
Faculty: Wydział Elektryczny
University: Politechnika Warszawska
PDF Hash:
15a93655ec5b86299add5308715b0e9a7071740709deac96cd29f2433d1e9298

Hash matches!

Generated hash: 15a93655ec5b86299add5308715b0e9a7071740709deac96cd29f2433d1e9298

Diploma Type: Inżynierski
Date of Defense: 2025-05-01
Supervisor: Robert Szmurło
Reviewer: Włodzimierz Dąbrowski

JWT Token

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVuaXZlcjNpdHkiOiJQb2pdGVjaG5pa2EgV2Fyc3phd3NrYSJ9.eyJwcmV2aW91c19ibG9jayl6IjMyOTk1NDlmY2lxYzY0YzlyMDc1

You can paste this token into the DPACHain Authenticator desktop app to re-verify its signature.

All Fields

Field	Value
Block ID	1
Timestamp	1745945769.053914
Diploma Type	Inżynierski
PDF Hash	15a93655ec5b86299add5308715b0e9a7071740709deac96cd29f2433d1e9298
Authors	Wojciech Szade
Authors IDs	319110
Title	Wykorzystanie technologii blockchain do podpisywania autentyczności prac dyplomowych
Language	PL
Discipline	Informatyka stosowana
Is Defended	0

Rysunek 14. Wyświetlone szczegóły dyplomu i zgodność załączonego pliku PDF

7.2 Dodawanie nowej pracy dyplomowej

Pracownik uczelni dodaje nową pracę za pomocą interfejsu dostępnego w panelu administracyjnym strony. Po wypełnieniu formularza i przesłaniu pliku PDF klikany jest przycisk **Create diploma**. System po walidacji dodaje blok do łańcucha i synchronizuje zmiany. Po poprawnym dodaniu nowej pracy wyświetlane są jej szczegóły.

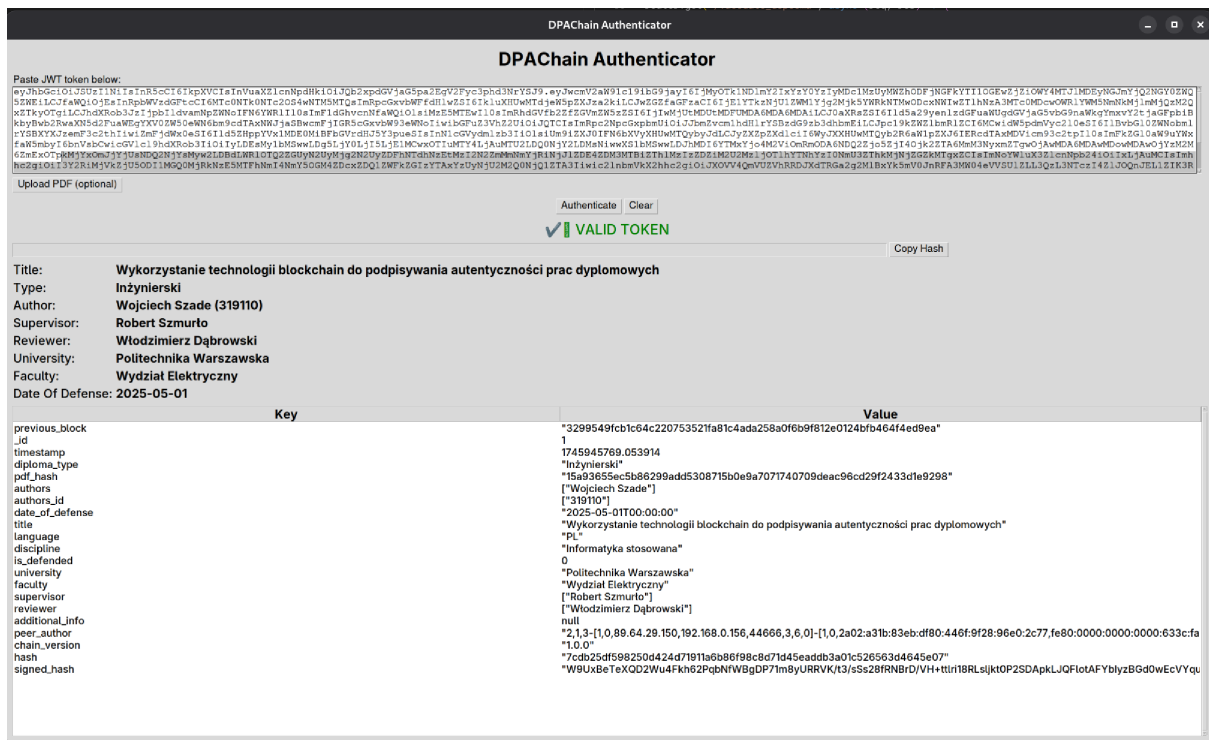
Widok tego interfejsu przedstawiono na rysunku 16.

7.3 Instrukcja obsługi systemu dla administratorów

7.3.1 Konfiguracja wstępna

Administrator przygotowujący węzeł pobiera kod źródłowy systemu i instaluje wymagane narzędzia: Docker oraz interpretator Python z biblioteką *cryptography*. Następnie generuje parę kluczy RSA za pomocą skryptu `generate_signing_keys.py`.

Rozdział 7. Instrukcja korzystania z systemu



Rysunek 15. Aplikacja do potwierdzania autentyczności danych

DPACHain API Manager Admin Staff User

Create diploma

Diploma Type* Inzynierski Title* 'korzystanie technologii blockchain do podpisywania autentyczności prac dyplomowych'

Language* PL Discipline* Informatyka stosowana Defended? (0/1)* 0

Date of Defense* 05 / 01 / 2025 University* Politechnika Warszawska

Faculty* Wydział Elektryczny Additional Info

Authors* Wojciech Szade Authors IDs* 319110 Supervisor* Robert Szmurło

Reviewer* Włodzimierz Dąbrowski

PDF File* Przeglądaj... Projekt_dyplomowy_2_0.pdf

Create diploma

Rysunek 16. Dodawanie nowej pracy do systemu – formularz

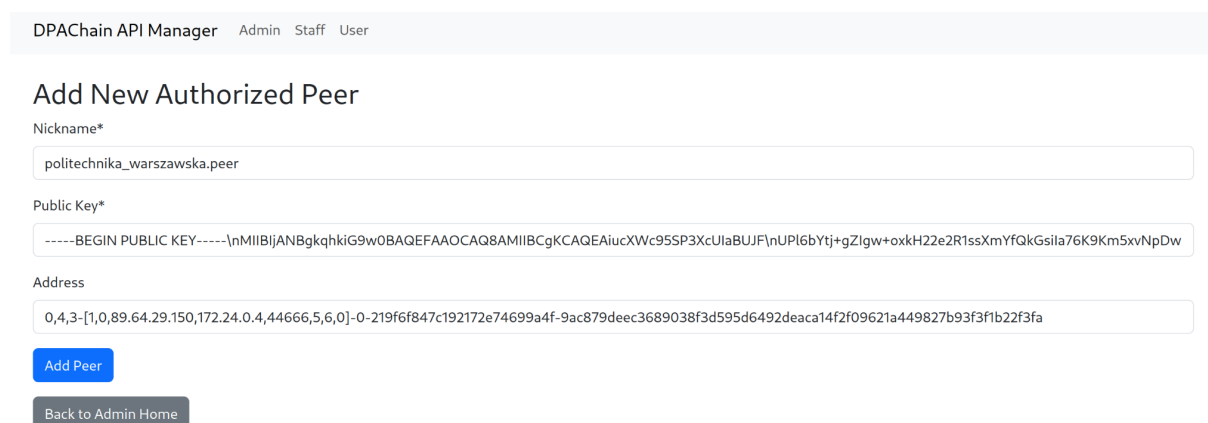
Po wprowadzeniu danych konfiguracyjnych do pliku `.env`, uruchamia system za pomocą komendy `docker compose up -d`. Interfejs API dostępny jest na porcie 8000, natomiast panel zarządzania na porcie 3000.

Po zalogowaniu do panelu administratora użytkownik powinien przejść do konfiguracji listy węzłów – aby węzeł poprawnie uczestniczył w sieci, konieczne jest dodanie do niego innych autoryzowanych węzłów, z którymi będzie mógł synchronizować łańcuch bloków.

7.3.2 Zarządzanie węzłami

Administrator ma możliwość dodawania nowych węzłów poprzez podanie ich nazwy, klucza publicznego oraz (opcjonalnie) adresu IP. System wspiera użycie uproszczonych nazw DNS.

Interfejs dodawania nowego węzła przedstawiono na rysunku 17.



The screenshot shows the 'Add New Authorized Peer' form in the DPACHain API Manager. At the top, there is a navigation bar with 'DPACHain API Manager' and links for 'Admin', 'Staff', and 'User'. The form has three input fields: 'Nickname*' with the value 'politechnika_warszawska.peer', 'Public Key*' with a long alphanumeric string starting with '-----BEGIN PUBLIC KEY-----', and 'Address' with a long alphanumeric string. Below the fields are two buttons: a blue 'Add Peer' button and a grey 'Back to Admin Home' button.

Rysunek 17. Formularz dodawania nowego węzła

Widok listy węzłów systemu pokazano na rysunku 18.

Administrator może zarządzać węzłami – usuwać je, blokować, wysyłać zapytania o synchronizację oraz przedstawiać swój węzeł innym.

7.3.3 Synchronizacja łańcucha

Administrator ma możliwość ręcznego wywołania synchronizacji danych łańcucha za pomocą interfejsu przedstawionego na rysunku 19. Funkcja ta inicjuje proces porównywania łańcuchów i pobierania brakujących bloków od zaufanych węzłów.

DPACChain API Manager Admin Staff User

Peers List

Nickname	Address	Status	Authorized	Public Key	Actions
2,1,3- [1,0,89.64.29.150,192.168.0.156,44666,3,6,0]- [1,0,2a02:a31b:83eb:df80:446f:9f28:96e0:2c77,fe80:0000:0000:0000:633c:fa19:d261:bc5,44666,3,6,0]- de946de27e22867e2d1a57a71-3267ff2cfb4b62ed18d3710be8e323d6b3e639c99aa3ac246e7e8d23cdfd181d	2,1,3- [1,0,89.64.29.150,192.168.0.156,44666,3,6,0]- [1,0,2a02:a31b:83eb:df80:446f:9f28:96e0:2c77,fe80:0000:0000:0000:633c:fa19:d261:bc5,44666,3,6,0]- de946de27e22867e2d1a57a71-3267ff2cfb4b62ed18d3710be8e323d6b3e639c99aa3ac246e7e8d23cdfd181d	OWN	Yes	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlucXwc95SP3XclUP16bYtj+gZIGw+oxkH22e2R1ssXmyfQkGs1Ia76K9Km5xvNpDwma6HzYz hqswoxjEbnM/DsHPf+w2bu20h0QVn5vcHLSAnetPWV6IXInqMPCEoME4t tPAXKIEra0v9LP5KU5S5kdUTFDgDbdDU6q+/uOmkoEhf9ZJImTVo6exqNz JgrDbdNVUpeVcvtXbrseyQxN1C8YHuLFTH3tQInUpQ0ZTYD2FS+wtcdfAN/ 42jbdTgkYTBwBht92fsQ2bpVjJtHLSqNH/ZDXC5CZ2dD1qHFCALeL2dd7c nQIDAQAB -----END PUBLIC KEY-----	Details
politechnika_warszawska.peer	0,4,3- [1,0,89.64.29.150,172.24.0.4,44666,5,6,0]-0-219f6f847c192172e74699a4f-9ac879deec3689038f3d595d6492deaca14f2f09621a449827b93f3fb22f3fa	UNKNOWN	Yes	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlucXwc95SP3XclUP16bYtj+gZIGw+oxkH22e2R1ssXmyfQkGs1Ia76K9Km5xvNpDwma6HzYz hqswoxjEbnM/DsHPf+w2bu20h0QVn5vcHLSAnetPWV6IXInqMPCEoME4t tPAXKIEra0v9LP5KU5S5kdUTFDgDbdDU6q+/uOmkoEhf9ZJImTVo6exqNz JgrDbdNVUpeVcvtXbrseyQxN1C8YHuLFTH3tQInUpQ0ZTYD2FS+wtcdfAN/ 42jbdTgkYTBwBht92fsQ2bpVjJtHLSqNH/ZDXC5CZ2dD1qHFCALeL2dd7c nQIDAQAB -----END PUBLIC KEY-----	Details

Back to Admin Home

Rysunek 18. Lista węzłów w systemie

DPACChain API Manager Admin Staff User

Sync Chain

"Chain has been synced!"

Start Sync

Back to Admin Home

Rysunek 19. Panel do synchronizacji łańcucha

Rozdział 8

Podsumowanie

W niniejszej pracy zaproponowano i zrealizowano rozwiązanie problemu weryfikacji autentyczności prac dyplomowych z wykorzystaniem technologii blockchain. W odpowiedzi na zidentyfikowane ograniczenia istniejących systemów oraz brak jednolitego, niezależnego mechanizmu weryfikacji dokumentów akademickich, opracowano koncepcję i wykonano funkcjonalny prototyp systemu opartego na modelu zdecentralizowanym.

Przedstawione rozwiązanie łączy zalety trwałości, bezpieczeństwa i rozproszonej natury technologii blockchain, dostosowując je do potrzeb środowiska akademickiego poprzez zastosowanie konsensusu PoA. Dzięki temu system jest w stanie zapewnić niezmiennosć zapisanych danych, ich wiarygodność oraz odporność na manipulacje, przy minimalnych kosztach energetycznych i organizacyjnych.

8.1 Osiągnięte rezultaty

W ramach projektu stworzono kompletny zestaw komponentów systemu:

- backend odpowiedzialny za zarządzanie łańcuchem bloków, synchronizację sieciową i obsługę zapytań użytkowników,
- frontend webowy umożliwiający zarówno publiczną weryfikację prac dyplomowych przez pracodawców i absolwentów, jak i działania administracyjne, takie jak rejestracja nowych prac oraz zarządzanie listą węzłów,
- aplikację desktopową służącą do lokalnej weryfikacji tokenów weryfikacyjnych na podstawie podpisu cyfrowego uczelni.

Opracowana struktura implementuje mechanizm powiązań hashowych między blokami, podpisywanie bloków przy użyciu kluczy prywatnych węzłów oraz możliwość niezależnej weryfikacji autentyczności danych bez konieczności komunikacji z centralnym serwerem. Zastosowanie tokenów JWT umożliwia użytkownikom przeprowadzenie weryfikacji nawet w przypadku braku dostępu do sieci blockchain.

8.2 Napotkane ograniczenia i problemy

W trakcie realizacji projektu napotkano pewne wyzwania techniczne, związane głównie z ograniczeniami biblioteki *p2pd* obsługującej komunikację P2P. Ze względu na konieczność tworzenia nowych połączeń dla każdej wymiany wiadomości, wdrożenie stabilnego i wydajnego mechanizmu konsensusu zostało ograniczone do podstawowej jego implementacji, nie zawierającej mechanizmu wyboru lidera.

Ponadto obecna wersja systemu zakłada ręczne zarządzanie listą węzłów. W zamkniętym środowisku współpracujących uczelni nie stanowi to znaczącego ograniczenia, jednak w przypadku zmniejszenia liczby aktywnych autoryzowanych węzłów, taki sposób zarządzania mógłby utrudnić utrzymanie sieci.

Podczas analizy powstałego systemu zidentyfikowano istotne ograniczenie wynikające z decyzji projektowej, polegającej na zapisaniu wszystkich kluczy publicznych autoryzowanych węzłów bezpośrednio w bloku genezy. Rozwiązanie to miało na celu zapewnienie jednego, niezmiennego źródła prawdy – możliwego do odczytu przez każdy nowo dołączony węzeł bez potrzeby zaufania do zewnętrznych mechanizmów. Choć w ramach realizowanego systemu można przewidzieć i zarezerwować miejsce dla wszystkich potencjalnych uczestników (np. polskich uczelni) oraz przygotować dla nich odpowiednie klucze z wyprzedzeniem, rozwiązanie to pozostaje ograniczone i nieskalowalne w dłuższej perspektywie.

Kluczowym problemem tego podejścia jest brak możliwości unieważnienia raz opublikowanego klucza – np. w przypadku jego kompromitacji. System nie przewiduje dynamicznego zarządzania uprawnieniami, co oznacza, że nawet jeśli większość węzłów zdecyduje się ignorować podpisy pochodzące z danego źródła, formalnie nadal jest ono traktowane jako autoryzowane. W efekcie system nie spełnia w pełni przyjętego założenia dotyczącego możliwości skutecznego wykluczania węzłów, które utraciły integralność.

W przyszłości należałoby rozważyć rozdzielenie ról i przechowywanie w bloku genezy wyłącznie kluczy służących do weryfikacji podpisów tokenów JWT, lub całkowicie odejść od tego podejścia na rzecz źródła prawdy opartego na centralnej instytucji – tak jak jest to realizowane w przypadku pieczęci elektronicznej. Natomiast aktywną listę walidatorów można by utrzymywać w bardziej elastyczny sposób – np. poprzez mechanizm aktualizacji zatwierdzany przez większość uprawnionych węzłów – korzystając z PoA.

8.3 Możliwe usprawnienia i kierunki rozwoju

Wśród możliwych kierunków rozwoju systemu wyróżnić można:

- wdrożenie trwalszych połączeń sieciowych umożliwiających lepszą synchronizację i propagację bloków oraz pełną realizację algorytmu wyboru lidera odpowiedzialnego za generowanie nowych bloków,
- rozważenie alternatywnych metod weryfikacji dokumentów, takich jak kwalifikowana pieczęć elektroniczna, która mogłaby zastąpić obecnie stosowane tokeny JWT.

Zastosowanie kwalifikowanej pieczęci elektronicznej umożliwiłoby pełne uwierzytelnienie dokumentu bez konieczności posiadania dostępu do danych łańcucha bloków lub kluczy publicznych konkretnych uczelni. Byłoby to rozwiązanie w pełni zgodne z europejskimi regulacjami dotyczącymi usług zaufania (eIDAS), zwiększając formalną wartość prawną weryfikacji. Alternatywnie można rozważyć stworzenie hybrydowego systemu, w którym pieczęć elektroniczna oraz token JWT współistnieją, łącząc zalety obu podejść.

8.4 Wnioski końcowe

Realizacja projektu wykazała, że technologia blockchain, odpowiednio dostosowana do specyfiki środowiska akademickiego, może skutecznie rozwiązać problem trwałej, rozproszonej i bezpiecznej weryfikacji autentyczności prac dyplomowych. Opracowane rozwiązanie jest próbą stworzenia solidnej podstawy dla dalszych wdrożeń i rozwoju, wskazując jednocześnie na obszary, w których możliwe są przyszłe usprawnienia i optymalizacje. System zaproponowany w pracy może stanowić realne wsparcie dla instytucji edukacyjnych poszukujących nowoczesnych metod ochrony wiarygodności wydawanych przez siebie dokumentów.

Bibliografia

- [1] BigchainDB GmbH, „BigchainDB 2.0 The Blockchain Database”, BigchainDB GmbH, Berlin, Germany, spraw. tech., maj 2018, Paper version 1.0.
- [2] Chauhan, A., „A Review on Various Aspects of MongoDB Databases”, *International Journal of Engineering Research & Technology (IJERT)*, t. 8, nr. 05, maj 2019, IJERTV8IS050031. Published by: <http://www.ijert.org>. Licensed under a Creative Commons Attribution 4.0 International License., ISSN: 2278-0181.
- [3] European Commission, *Europass Digital Credentials Infrastructure*, Accessed: 2025-04-28, 2021. adr.: <https://ec.europa.eu/futurium/en/europass/europass-digital-credentials-infrastructure.html>.
- [4] Gdański, U., *Potwierdzanie autentyczności wykształcenia – kontakt mailowy*, <https://www.ug.edu.pl/weryfikacja-dyplomu>, 2024.
- [5] Golosova, J. i Romanovs, A., „The Advantages and Disadvantages of the Blockchain Technology”, w *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2018, s. 1–6. DOI: 10.1109/AIEEE.2018.8592253.
- [6] Grolleau, G., Lakhal, L. i Mzoughi, N., „An introduction to the economics of fake degrees”, *Journal of Economic Issues*, t. 42, nr. 3, s. 673–693, 2008.
- [7] Hayes, A. S., „A Cost of Production Model for Bitcoin”, Department of Economics The New School for Social Research, New York, NY, spraw. tech., lut. 2015.
- [8] John, K., O'Hara, M. i Saleh, F., „Bitcoin and Beyond”, *Annual Review of Financial Economics*, t. 14, s. 95–115, 2022.
- [9] nesdis, *Django Documentation*, Release 1.2.24, Read the Docs, kw. 2018. adr.: https://djangoapi.readthedocs.io/_/downloads/en/sphinx/pdf/.
- [10] Ongaro, D. i Ousterhout, J., „In search of an understandable consensus algorithm”, w *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, ser. USENIX ATC'14, Philadelphia, PA: USENIX Association, 2014, s. 305–320, ISBN: 9781931971102.
- [11] Politechnika Świętokrzyska, *Politechnika Świętokrzyska i Billon przenoszą dyplomy ukończenia uczelni na blockchain*, Informacja prasowa, PRNews.pl, lut. 2021. adr.: <https://prnews.pl/politechnika-swietokrzyska-i-billon-przenosza-dyplomy-ukonczenia-uczelni-na-blockchain-456648>.

- [12] Raikwar, M., Gligoroski, D. i Krlevska, K., „SoK of Used Cryptography in Blockchain”, *IEEE Access*, t. 7, s. 1–1, paź. 2019. DOI: 10.1109/ACCESS.2019.2946983.
- [13] Saad, M., Spaulding, J., Njilla, L., Kamhoua, C. A., Nyang, D. i Mohaisen, A., „Overview of attack surfaces in blockchain”, w *Blockchain for Distributed Systems Security*, Shetty, S. S., Kamhoua, C. A. i Njilla, L. L., red., Wiley-IEEE Computer Society Pr, 2019, s. 51–62, ISBN: 978-1-119-51960-7.
- [14] Sayed, R. H., „Potential of Blockchain Technology to Solve Fake Diploma Problem”, prac. mag., University of Jyväskylä, Department of Computer Science i Information Systems, 2019.
- [15] Segendorf, B., „Have virtual currencies affected the retail payments market?”, *Economic Commentaries*, no. 2, Sveriges Riksbank, 2014.
- [16] Szkoła Główna Handlowa w Warszawie, *Przypieczętowane. Dyplomy absolwentów SGH z pieczęcią elektroniczną*, *Gazeta SGH* (online), 2020. adr.: <https://gazeta.sgh.waw.pl/aktualnosci/przypieczetowane-dyplomy-absolwentow-sgh-z-pieczecia-elektroniczna>.
- [17] Turing, *Python FastAPI vs Flask: A Detailed Comparison*. adr.: <https://www.turing.com/kb/fastapi-vs-flask-a-detailed-comparison>.
- [18] w Polsce, P. N., *Kielce/ Cyfrowe dyplomy na Politechnice Świętokrzyskiej dzięki blockchain*, Accessed: 2025-04-28, 2021. adr.: <https://naukawpolsce.pl/aktualnosci/news%2C86214%2Ckielce-cyfrowe-dyplomy-na-politechnice-swietokrzyskiej-dzieki-blockchain>.
- [19] w Warszawie, S. G. H., *Elektroniczna weryfikacja dyplomu*, <https://www.sgh.waw.pl/pl/uczelnia/struktura/pieczec-elektroniczna>, 2024.

Wykaz skrótów i symboli

EDCI Europass Digital Credentials Infrastructure 11

JCS JSON Canonicalization Scheme 35

JWT JSON Web Token 3, 22, 23, 26, 29–31, 35, 44, 49–51

P2P Peer to Peer 8, 31, 36, 40, 50

PKI Public Key Infrastructure 13, 15

PoA Proof of Authority 3, 22, 23, 49, 50

PŚ Politechnika Świętokrzyska 14

SGH Szkoła Główna Handlowa w Warszawie 13–15

Słownik pojęć

bitcoin pierwsza i najpopularniejsza kryptowaluta oparta na technologii blockchain 15

blockchain technologia rozproszonego rejestru umożliwiająca przechowywanie niezmiennych zapisów transakcji 13–15

blok genezy Pierwszy blok łańcucha, który zawiera inną zawartość niż pozostałe bloki. Blok genezy nie zawiera też informacji o bloku poprzedzającym go – ponieważ jest pierwszy. 22

eIDAS rozporządzenie Unii Europejskiej regulujące identyfikację elektroniczną i usługi zaufania 13, 57

hash wynik działania funkcji haszującej – przyporządkowanie dowolnych danych wejściowych do hasha o stałej długości, w taki sposób, że te same dane zawsze generują ten sam hash, a różne dane – z bardzo wysokim prawdopodobieństwem – generują różne hashe 31, 44

JSON Canonicalization Scheme standard serializacji danych JSON, który zapewnia jednoznaczność i uporządkowaną strukturę poprzez ustalone sortowanie kluczy i formatowanie danych. Umożliwia tworzenie spójnych podpisów cyfrowych wiadomości JSON 35, 55

JSON Web Token standard przesyłania informacji między stronami w formacie JSON, zawierający podpis cyfrowy umożliwiający weryfikację integralności i autentyczności danych 3, 55

kryptograficzny odcisk dokumentu unikalny skrót kryptograficzny (hash) pracy dyplomowej, wykorzystywany do jej weryfikacji 17, 43

kwalifikowana pieczęć elektroniczna pieczęć elektroniczna przypisana do instytucji, zapewniająca integralność i autentyczność dokumentów zgodnie z regulacjami eIDAS 13

Peer to Peer Sieć w której uczestnicy komunikują się bezpośrednio między sobą, bez potrzeby centralnego serwera 8, 55

Proof of Authority mechanizm konsensusu w sieciach blockchain, w którym nowe bloki mogą dodawać jedynie autoryzowane węzły identyfikowane za pomocą kluczy publicznych 3, 55

Public Key Infrastructure infrastruktura klucza publicznego umożliwiająca bezpieczne uwierzytelnianie i podpisywanie danych 13, 55

Raft algorytm konsensusu umożliwiający utrzymanie spójności danych w rozproszonym systemie poprzez wybór lidera 16

Tendermint protokół komunikacji sieciowej i mechanizm konsensusu wykorzystywany w zdecentralizowanych systemach 16

Spis rysunków

1	Diagram przypadków użycia studenta	18
2	Diagram przypadków użycia pracodawcy	18
3	Diagram przypadków użycia pracownika uczelni	19
4	Diagram przypadków użycia administratora	19
5	Diagram komponentów architektury systemu	27
6	Diagram architektury fizycznej systemu	28
7	Proces podpisywania bloku [12]	29
8	Diagram klasy bloku	30
9	Diagram klasy bloku genezy	31
10	Diagram sekwencji procesu uruchamiania aplikacji	33
11	Diagram sekwencji procesu dodawania nowego bloku	34
12	Strona do weryfikacji dyplomu	44
13	Komunikat o nieistnieniu żądanej pracy	44
14	Wyświetlone szczegóły dyplomu i zgodność załączonego pliku PDF	45
15	Aplikacja do potwierdzania autentyczności danych	46
16	Dodawanie nowej pracy do systemu – formularz	46
17	Formularz dodawania nowego węzła	47
18	Lista węzłów w systemie	48
19	Panel do synchronizacji łańcucha	48

Spis tabel

1	Porównanie bibliotek FastAPI i Flask [17]	40
---	---	----

Spis załączników

1. Link do repozytorium zawierającego opisywany projekt
<https://github.com/WojciechSzade/dpachain>