

**Piotr Filipkowski**

Szkoła Główna Handlowa w Warszawie  
e-mail: pfilip@sgh.waw.pl

**Michał Horodelski**

Katolicki Uniwersytet Lubelski Jana Pawła II  
e-mail: michal.horodelski@kul.lublin.pl

**Krystyna Polańska**

Szkoła Główna Handlowa w Warszawie  
e-mail: kpolan@sgh.waw.pl

---

***BLOCKCHAIN W ZDECENTRALIZOWANEJ  
AUTORYZACJI TRANSAKCJI BARTEROWYCH***  
***BLOCKCHAIN IN DECENTRALIZED  
AUTHORIZATION OF BARTER TRANSACTIONS***

---

DOI: 10.15611/pn.2018.527.06  
JEL Classification: O14, O39, L14

**Streszczenie:** Celem artykułu jest przedstawienie koncepcji modelu systemu wymiany barterowej realizującego autoryzację transakcji za pomocą łańcucha bloków (*blockchain*). Współczesne tendencje rozwoju działalności biznesowej wiodące w kierunku gospodarki współdzielenia (*sharing economy*) wskazują, że transakcje bez ekwiwalentu pieniężnego będą stosowane coraz częściej. Wymiana barterowa, czyli towar/usługa za towar/usługę wpisuje się w ten nurt transakcji gospodarczych. Różnorodność i liczba takich transakcji będzie wzrastać wraz ze wzrostem możliwości obliczeniowych superkomputerów niezbędnych do obsługi tego typu wymiany gospodarczej. Problem w tym, jak skutecznie zabezpieczyć systemy wymiany barterowej np. przed wielokrotnym wykorzystaniem tych samych towarów/usług w wielu różnych transakcjach toczących się równocześnie. W artykule zaproponowano model systemu wymiany barterowej, w którym autoryzacja transakcji wykorzystuje koncepcję rozproszonej bazy danych.

**Słowa kluczowe:** łańcuch bloków, transakcje barterowe, autoryzacja transakcji.

**Summary:** The aim of the article is to present the concept of model of the system performing authentication barter transactions using a blockchain. Modern trends of business development leading to sharing economy indicate that transactions without cash equivalent will be used more and more often. Barter exchange is part of this trend of economic transactions. The variety and number of such transactions will increase with the increase of computing capabilities of supercomputers necessary to support this type of business exchange. The problem is how

to effectively secure barter exchange systems, for example, against the multiple use of the same goods/services in many different transactions taking place simultaneously. The article proposes a model of a barter exchange system in which transaction authorization uses the concept of a distributed database.

**Keywords:** blockchain, barter transaction, authorization of transaction.

## 1. Wstęp

Problem efektywnej wymiany towar za towar ma swoje źródła już w starożytności. System barterowy jest dobrze znanym i zdefiniowanym bezpiecznym sposobem wymiany towar za towar [Demirkol i in. 2011].

Współcześnie obserwuje się tendencje rozwoju działalności biznesowej z obszaru gospodarki współdzielenia (*sharing economy*), a szczególnie te, które wykonują transakcje bez ekwiwalentu pieniężnego. Wymiana barterowa, czyli towar/usługa za towar/usługę, wpisuje się w ten nurt transakcji gospodarczych. Różnorodność i liczba takich transakcji będzie wzrastać wraz ze wzrostem możliwości obliczeniowych superkomputerów niezbędnych do obsługi tego typu wymiany gospodarczej.

Aktualnie systemy barterowe są też coraz częściej stosowane jako narzędzie marketingowe. Firmy upłynniają niesprzedane towary lub niedostatecznie wykorzystywane usługi na rynkach barterowych. Do wsparcia tego typu działalności na rynkach elektronicznych wykorzystuje się rozwiązania programowe w postaci agentów pośredniczących, ponieważ częste przeszukiwanie różnych baz danych, a tym bardziej negocjowanie warunków wymiany może być uciążliwe.

Znane są w literaturze rozwiązania mające usprawnić proces wymiany barterowej, m.in. poprzez zastosowanie ontologii jako rozwiązania problemu wymiany towarów lub usług na inne bez użycia pieniądza. Problem przetargowy rozwiązywany jest za pośrednictwem autonomicznych agentów współpracujących z agentem *matchmaker*, którego głównym celem jest znalezienie odpowiedniej oferty barterowej dla swojego klienta. Wyszukanie drugiej strony kontraktu bez wykorzystania możliwości obliczeniowych agenta pośredniczącego nie jest proste ze względu na dynamicznie zmieniające się warunki wymiany na rynku dóbr lub usług [Cakmaz i in. 2017].

Celem artykułu jest przedstawienie koncepcji zabezpieczenia transakcji barterowych oraz ocena jej implementacji.

## 2. Systemy barterowe a łańcuch bloków – przegląd literatury

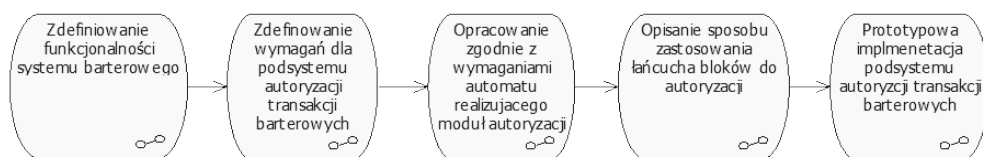
*Blockchain*, tłumaczony na język polski jako łańcuch bloków lub zdecentralizowana i rozproszona baza danych, może być wykorzystywany w różnych obszarach działalności gospodarczej. Obecnie *blockchain* wykorzystywany jest m.in. w: zaufanym

systemie śledzenia przesyłek i transportów obejmującym działaniem wielu niezależnych operatorów [Dickson 2016], obsłudze rywalizacji w grach on-line bez zaufanej instytucji obsługującej rankingi czy systemy oceny treści w portalach społecznościowych [Decentralized eSports Platform... 2016], prowadzeniu rejestrów własności nieruchomości [Ubitquity 2018], identyfikacji diamentów i rejestrowaniu ich transakcji [Fortune 2017], rejestrowaniu transakcji w systemach elektroenergetycznych [Riva Sanseverino i in. 2017], uniwersalnych platformach zarządzania tożsamością [Fortune 2018], rozwiązaniach wspierających procesy administracyjne i prawne [McMullen 2017] czy prowadzeniu medycznej dokumentacji pacjentów [Forbes 2017]. Ostatnio do najpopularniejszych należy jednak zaliczyć zastosowania w sektorze finansowym. Według badania branżowego przeprowadzonego przez Accenture aż 9 z 10 managerów przyznało, że reprezentowane przez nich banki aktywnie interesują się potencjalnymi zastosowaniami łańcucha bloków [Accenture 2018]. Do najciekawszych możliwych zastosowań należy zaliczyć m.in.: realizację bezpośrednich rozliczeń pomiędzy dwoma dowolnymi bankami [IBM 2017], wiarygodne rejestry polis ubezpieczeniowych [PwC 2017], zdecentralizowany system obrotu papierami wartościowymi [Deloitte 2016] czy kryptowaluty [Narayanan i in. 2016].

Dokonując szerokiej kwerendy literaturowej autorzy nie odnaleźli publikacji z obszaru zastosowania *blockchain* do zabezpieczenia rejestru transakcji barterowych. Fakt ten uzasadnia cel pracy, w ramach którego przedstawiono model systemu wymiany barterowej realizującego autoryzację transakcji zgodnie z przyjętymi kryteriami. Problem badawczy pracy dotyczy więc adaptacji łańcucha bloków do realizacji zdecentralizowanego podsystemu autoryzacji transakcji wymiany towar/usługą za towar/usługę.

### 3. Modelowanie systemu wymiany barterowej

W związku ze zidentyfikowanym problemem autorzy pracy przy wsparciu języka UML proponują metodę modelowania podsystemu zdecentralizowanej autoryzacji historii transakcji, w możliwie prosty sposób, wg schematu na rys. 1.



**Rys. 1.** Schemat metody modelowania (wytworzenia) modułu autoryzacji w systemie barterowym

Źródło: opracowanie własne.

Proces modelowania rozpoczyna się od identyfikacji funkcji systemu barterowego, który może zawierać podsystem zdecentralizowanej autoryzacji historii transakcji. Następnie odbywa się doprecyzowanie wymagań stawianych przypadkom użycia systemu oraz przedstawienie założeń działania systemu. W kolejnym etapie zostaje opracowany model automatu reprezentujący system barterowy w aspekcie autoryzacji transakcji. Podkreślono tu sposób zastosowania łańcucha bloków do autoryzacji historii transakcji. Ostatnim etapem jest prototypowa implementacja modułu autoryzacji osadzona w systemie barterowym.

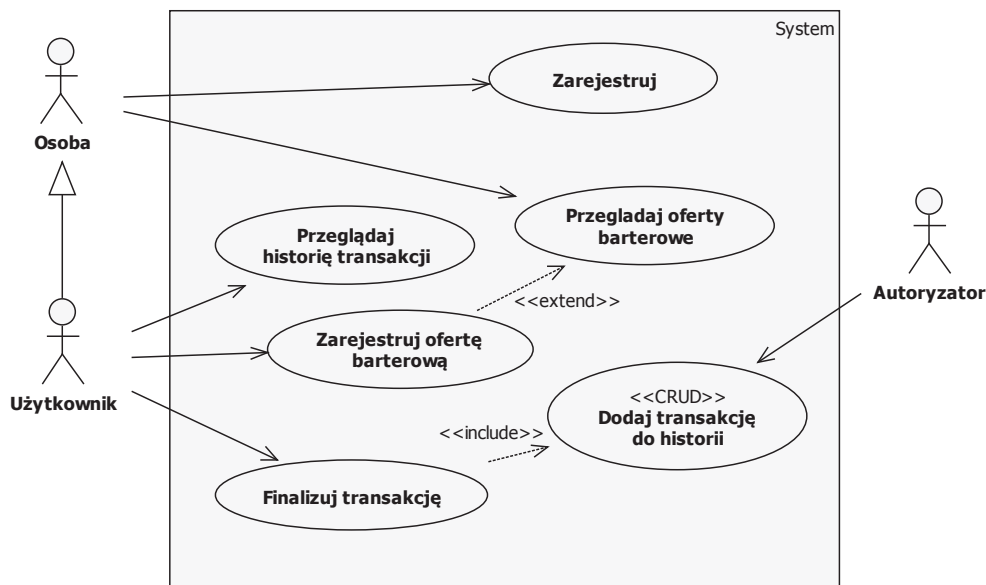
### 3.1. Funkcjonowanie systemu barterowego

Proces modelowania rozpoczyna zdefiniowanie nieformalnego opisu kluczowych wymagań systemu barterowego. System informatyczny wspomagający wymianę barterową powinien tworzyć platformę umożliwiającą: przede wszystkim dodawanie ofert barterowych wymiany towar/usługa za towar/usługę, kojarzenie dodanych ogłoszeń, optymalizując korzyść dla całej społeczności użytkowników platformy, a gdy nastąpi zgodność zgłoszonych potrzeb – finalizowanie transakcji wymiany barterowej. W przypadku wymiany bezgotówkowej nie jest wymagana trzecia strona autoryzująca transakcję pieniężną. Wystarczającą autoryzacją transakcji jest więc nadzór społeczności użytkowników systemu. W związku z powyższym system powinien zawierać możliwość automatyzacji autoryzacji przeprowadzonych transakcji, a odpowiedzialność za poprawność przechowywania historii transakcji powinna spoczywać na wszystkich użytkownikach systemu. Automatyzacja autoryzacji jest konieczna z powodu rosnącej ilości danych.

Podstawowe funkcje systemu przedstawiono na diagramie przypadków użycia UML na rys. 2. System posiada trzech aktorów społecznych. Aktor *Użytkownik* jest osobą, która ma konto w systemie i ma prawo do dodawania ofert barterowych, składania propozycji barterowych oraz finalizowania transakcji barterowych. Aktor *Osoba* może jedynie przeglądać transakcję, choć może także ubiegać się o założenie konta w systemie. Aktor *Autoryzator* jest podprogramem realizującym autoryzację transakcji metodą łańcucha bloków.

Z punktu widzenia niniejszej pracy najistotniejszą funkcją systemu jest *Dodaj transakcję do historii* nawigowaną przez *Autoryzatora* oraz *Przełóż historię transakcji* nawigowaną przez *Użytkownika*.

Każdy z przypadków użycia można zdefiniować przy użyciu tabelki charakteryzującej czynność. W tabeli 1 skupiono się na scenariuszach głównych przypadków użycia.



Rys. 2. Diagram przypadków użycia systemu barterowego

Źródło: opracowanie własne.

Tabela 1. Zestawienie scenariuszy głównych przypadków użycia systemu barterowego

Nazwa przypadku użycia	Scenariusz główny
Przeglądaj oferty barterowe	<i>Użytkownik</i> , korzystając z możliwości filtrowania danych, przegląda dostępne w systemie oferty.
Zarejestruj ofertę barterową	<i>Użytkownik</i> uruchamia formularz, w którym definiuje zbywany towar/usługę, po czym zatwierdza dodanie do systemu.
Finalizuj transakcję	<b>Przypadek 1.</b> Transakcja zaakceptowana przez jednego ze zbywających: Jeden ze zbywających zaznacza skojarzoną przez system transakcję jako możliwą do finalizacji, a następnie oczekuje na rezultat ze strony drugiego zbywającego. <b>Przypadek 2.</b> Transakcja zaakceptowana przez obydwie strony. Jeden ze zbywających zaznacza skojarzoną przez system transakcję jako do finalizacji, co uruchamia proces autoryzacji transakcji i zapisu w historii.
Dodaj transakcję do historii	<i>Autoryzator</i> synchronicznie z <i>Użytkownikiem</i> uruchamia akcję zwaną <i>rozpocznij transakcję</i> . Następnie transakcja zostaje zdefiniowana, zabezpieczona wygenerowanym kluczem oraz zapisana do bazy danych.
Przeglądaj historię transakcji	<i>Użytkownik</i> uruchamia przeglądarkę transakcji przeprowadzonych w systemie w celu odnalezienia konkretnych informacji. W razie potrzeby <i>Użytkownik</i> uruchamia weryfikację poprawności wyświetlanych transakcji przy pomocy <i>Autoryzatora</i> .

Źródło: opracowanie własne.

### 3.2. Wymagania podsystemu autoryzacji transakcji barterowych

Moduł autoryzacji jest umiejscowiony w funkcji *dodania transakcji do historii*, w bazie danych systemu. *Autoryzator* synchronicznie z *Użytkownikiem* uruchamia akcję zwaną *rozpocznij transakcję*. Następnie transakcja zostaje zdefiniowana, zabezpieczona wygenerowanym kluczem oraz zapisana do bazy danych.

W przypadku tak zaprojektowanych funkcji systemu barterowego rozważana jest sytuacja, w której modelowany system z zawartym modułem autoryzacji będzie spełniał założenia wymienione w specyfikacji:

- Każdy *Użytkownik* systemu zawsze może zarejestrować w pewnym momencie działania systemu nową ofertę barterową.
- Każdy *Użytkownik* (zbywający nr 1) w pewnym momencie może sfinalizować wyznaczoną przez system transakcję barterową jeżeli tylko drugi *Użytkownik* (zbywający nr 2) też chce ją sfinalizować.
- Jeżeli *Użytkownik* rozpoczął finalizację transakcji, to *Autoryzator* doda ją do historii i zabezpieczy wygenerowanym kluczem.
- Zmiana danych w przeprowadzonych transakcjach nie jest możliwa, aczkolwiek możliwe jest przeprowadzenie odwróconej transakcji.
- System w bazie danych przechowuje dane *Użytkowników* (zbywających), dane towarów/usług oraz historie transakcji.
- Transakcja barterowa zawiera dane: data zawarcia umowy, dane zbywającego nr 1, dane zbywającego nr 2, dane towaru/usługi zbywającego nr 1, dane towaru/usługi zbywającego nr 2 oraz kod zabezpieczający.
- System na żądanie *Użytkownika* może wyświetlić historie transakcji wraz z weryfikacją ich poprawności.
- *Autoryzator* tuż przed dodawaniem nowej transakcji do systemu blokuje działanie innych *Autoryzatorów*, aby w tym samym czasie nie dodały innej transakcji powodującej konflikt interesów.

### 3.3. Automat realizujący moduł autoryzacji

System wymiany barterowej, w którym osadzony jest moduł autoryzacji, można rozważać jako system wieloagentowy. Agentami systemu są *Użytkownicy* i *Autoryzatorzy*, których współdziałanie można modelować superpozycją automatów ich reprezentujących.

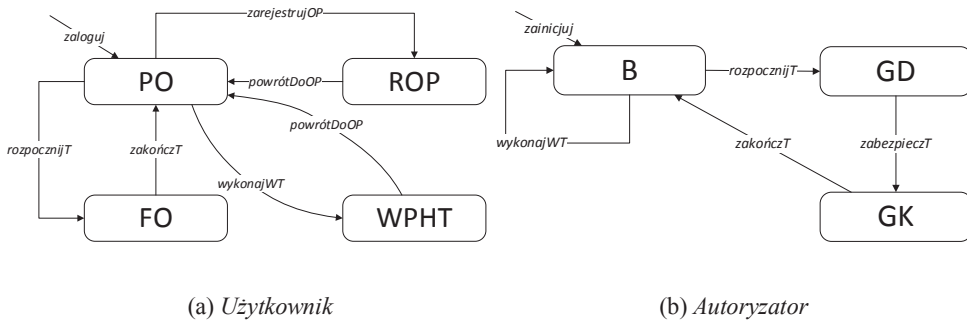
A zatem *i*-ty *Użytkownik*, którego automat  $U_i$  zaprezentowano na rys. 3a, może być w następujących stanach:

- przeglądanie ofert (stan początkowy) – PO,
- rejestrowanie oferty lub propozycji – ROP,
- finalizowanie oferty – FO,
- weryfikowanie przechowywanej historii transakcji – WPHT.

Natomiast akcje, jakie może wykonać, to:

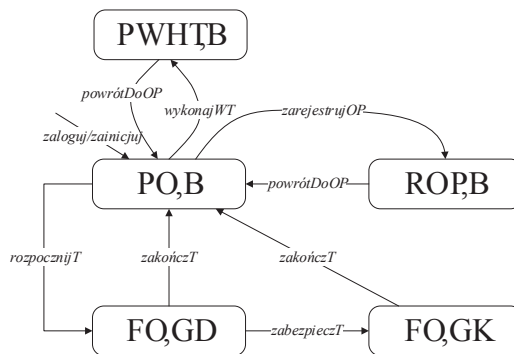
- zalogowanie do systemu – *zaloguj*,

- zarejestruj ofertę lub propozycję – *zarejestrujOP*,
- wróć do przeglądania ofert – *powrótDoPO*,
- rozpocznij transakcję – *rozpocznijT*,
- zakończ transakcję – *zakończT*,
- sprawdź zweryfikowaną historię transakcji – *wykonajWT*.



**Rys. 3.** Automat modelujący zachowanie agentów *Użytkownik* i *Autoryzator*

Źródło: opracowanie własne.



**Rys. 4.** Automat modelujący system barterowy  $SB_i = (U_i, A_i)$

Źródło: opracowanie własne.

Natomiast *i*-ty *Automat*  $A_i$  (odpowiadający *i*-temu *Użytkownikowi*) reprezentujący zachowanie *Autoryzatora*, prezentowany na rys. 3b, może być w następujących stanach:

- bezczynność (stan początkowy) – B,
  - generowanie danych transakcji – GD,
  - generowanie kodu zabezpieczającego – GK
- oraz może wykonać akcje:
- zainicjuj działanie – *zainicjuj*,

- rozpocznij transakcję – *rozpocznijT*,
- zabezpiecz transakcję – *zabezpieczT*,
- zapisz i zakończ transakcję – *zakończT*,
- wykonaj weryfikację historii transakcji – *wykonajWT*.

Agenci: *Użytkownik*  $U_i$  i *Autoryzator*  $A_i$  synchronizują się, wykonując akcję rozpoczęcia transakcji (*rozpocznijT*), akcję zakończenia transakcji (*zakończT*) oraz akcję weryfikacji historii transakcji (*wykonajWT*), co przedstawiono na rys. 4.

System barterowy w rezultacie jest modelowany przez superpozycję automatów modelujących zachowanie *Użytkownika* i *Autoryzatora*.

#### 4. Zabezpieczanie transakcji

Każdej zapisanej transakcji w tabeli TRANSACTION w bazie danych systemu przypisano określony numer transID. Podczas autoryzacji *Autoryzator* każdą transakcję zabezpiecza kluczem. Załóżmy, że transakcje w systemie są numerowane od 0 (transID). W celu zabezpieczenia transakcji kluczem wygenerowanym potrzebny jest klucz zabezpieczający poprzednio autoryzowanej transakcji. Jeżeli w systemie nie autoryzowano jeszcze żadnej transakcji (stan początkowy istnienia systemu), to wykorzystywany jest jedyny raz klucz startowy. Jeżeli *Autoryzator* zautoryzuje  $N-1$  nowych transakcji, to dodatkowo przeprowadza operację zamknięcia tych transakcji w blok, generując pustą transakcję autoryzowaną zabezpieczającą dany blok. Pozwala to na zmniejszenie przekazywanych danych historycznych przy autoryzacji, a nie wyklucza uwzględnienia żadnych transakcji z historii przy generowaniu kolejnego klucza zabezpieczającego.

Zastosowano pseudokod (przy założeniu, że w bloku jest zamykanych  $N-1$  transakcji):

1. *Autoryzator* przyjmuje zgłoszenie zabezpieczenia kolejnej transakcji o numerze  $J$  oraz blokuje możliwość autoryzowania przez innego *Autoryzatora*.

2. *Autoryzator* generuje klucz zabezpieczający transakcję  $J$ -tą przy wykorzystaniu klucza transakcji  $J-1$  oraz zapisuje transakcję w bazie.

3. Jeżeli reszta z dzielenia  $J$  przez  $N$  daje  $N-2$ , to:

3.1 *Autoryzator* definiuje pustą transakcję o numerze  $J+1$

3.2 *Autoryzator* generuje klucz zabezpieczający transakcję  $J+1$ -szą przy wykorzystaniu zbioru  $N-1$  poprzednich kluczy transakcji o numerach  $J-(N-2)$ , ...,  $J-1$ ,  $J$  oraz zapisuje transakcję w bazie.

4. *Autoryzator* odblokowuje możliwość autoryzowania przez innego *Autoryzatora*.

Poniżej zaprezentowano schematyczną postać tabeli TRANSACTION (tab. 3) z bazy danych systemu barterowego rozszerzoną o kolumny ułatwiające sprawdzenie punktu 3 z powyższego pseudokodu.



**Tabela 3.** Schematyczna postać tabeli TRANSACTION

transID	Inne kolumny transakcji	Generowanie klucza (zależnie od)	Wynik dzielenia	Warunek z p. 3
0	szczegóły transakcji nr 0	generujKlucz(pusty)	$0\%N==0$	–
1	szczegóły transakcji nr 1	generujKlucz(transID(0))	$1\%N==1$	–
...	...	...	...	...
$J'==N-2$	szczegóły transakcji nr $N-2$	generujKlucz(transID( $N-3$ )))	$(N-2)\%N==N-2$	spełniony
$N-1$	pusta transakcja nr $N-1$	generujKlucz(transID( $0:(N-2)$ )))	$(N-1)\%N==N-1$	–
$N$	szczegóły transakcji nr $N$	generujKlucz(transID( $N-1$ )))	$N\%N==0$	–
...	...	...	...	...
$M$	szczegóły transakcji nr $M$	generujKlucz(transID( $M-1$ )))	$M\%N==0$	–
$M+1$	szczegóły transakcji nr $M+1$	generujKlucz(transID( $M$ )))	$(M+1)\%N==1$	–
...	...	...	...	...
$J''==M+N-2$	szczegóły transakcji nr $M+N-2$	generujKlucz(transID( $M+N-3$ )))	$(M+N-2)\%N==N-2$	spełniony
$M+N-1$	pusta transakcja nr $M+N-1$	generujKlucz(transID( $M:(M+N-2)$ )))	$(M+N-1)\%N==N-1$	–
$M+N$	szczegóły transakcji nr $M+N$	generujKlucz(transID( $M+N-1$ )))	$(M+N)\%N==0$	–
...	...	...	...	...

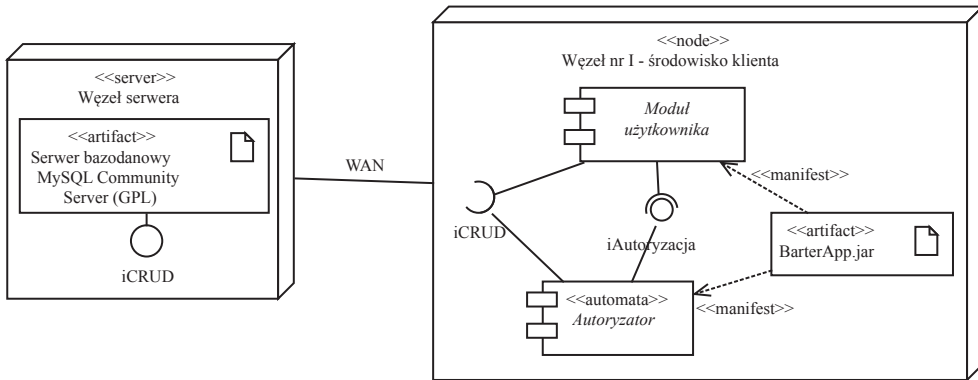
Źródło: opracowanie własne.

Zakładamy, że  $M > 0$  oraz że w bazie znajduje się co najmniej  $M+N+1$  rekordów transakcji. Natomiast funkcja *generujKlucz(dane)* generuje klucz zabezpieczający na podstawie danych *dane*. *transID(i)* oznacza dane *i*-tej transakcji. Symbol „%” reprezentuje dzielenie całkowite z resztą.

Przykładowe transakcje o numerach  $J'$  i  $J''$  są ostatnimi faktycznymi transakcjami w swoim bloku. Transakcja  $J'$  w bloku transakcji o numerach od 0 do  $(N-1)$ , a transakcja  $J''$  w bloku od  $M$  do  $M+N-1$ . Transakcje puste o numerach  $J'+1$  i  $J''+1$  zamykają blok. Natomiast transakcje o numerach  $J'+2$  i  $J''+2$  są pierwszymi faktycznymi transakcjami w kolejnych blokach, które są przechowywane w bazie danych systemu. Tak zabezpieczone dane w tabeli TRANSACTION nie mogą ulec zmianie, gdyż spowoduje to zasygnalizowanie niezgodności w widoku przeglądania historii transakcji.

## 5. Implementacja podsystemu autoryzacji transakcji barterowych

Struktura systemu wymiany barterowej, zaprojektowana w celu pokazania wdrożenia i oceny pomysłu zdecentralizowanej autoryzacji historii transakcji, może zostać przedstawiona na diagramie wdrożeniowym UML.



**Rys. 5.** Struktura prototypowego systemu barterowego dla  $I = 1, 2, \dots, n$ , (gdzie  $n$  to liczba użytkowników)

Źródło: opracowanie własne.

System składa się z węzła serwerowego przechowującego bazę danych oraz z węzłów środowiska klienckiego, w którym operują *Użytkownicy* systemu. Węzeł serwera udostępnia zestaw operacji *iCRUD* pozwalający na operowanie bazą danych. *I*-ty węzeł środowiska klienta zawiera dwa komponenty: *Moduł użytkownika* oraz *Autoryzator*. Komponent *Moduł użytkownika* zawiera funkcje związane z przeglądaniem ofert, rejestrowaniem ofert, składaniem propozycji transakcji oraz finalizacją transakcji. Komponent *Autoryzator* realizuje proces autoryzacji poprzez zabezpieczenie transakcji kluczem i zapisaniem jej do bazy danych, a także weryfikuje wyświetlenie historii transakcji. Oba komponenty są implementowane przez program oznaczony symbolicznie *BarterAPP.jar*.

Do implementacji prototypowej wykorzystano:

- serwer bazy danych: MySQL,
- połączenie z bazą danych: Java JDBC,
- program testowy: Aplikacja Java,
- funkcję skrótu: SHA-256.

Zadaniem prototypowej implementacji jest przetestowanie i ocena przede wszystkim mechanizmu zabezpieczania transakcji. W tym celu zaimplementowano działanie *Autoryzatora* zgodnie z pseudokodem, dla którego zmierzono czas dodawania nowej zabezpieczonej transakcji. Rozważono dwa scenariusze: dodanie

nowej transakcji bez zamykania w blok (NT) (średni czas 306,5 ms) oraz dodanie nowej transakcji wraz z zamknięciem w blok N-1 transakcji (NT+Z) (średni czas 328,1 ms). Testy wykonano na komputerze z procesorem i7 i 8 GB pamięci RAM. Liczba rekordów w tabeli TRANSACTION była nie większa niż 200, a liczba zamykanych w blok transakcji to 23.

Prototypowa implementacja wykazała, że system barterowy z modulem zdecentralizowanej autoryzacji historii transakcji funkcjonuje zgodnie z przyjętymi założeniami w racjonalnym czasie w kontekście realizacji transakcji barterowych.

Korzyścią dla użytkowników takiego systemu jest po pierwsze, pewność, że w tym samym czasie zbywcy uczestniczący w transakcji nie oferują zbywanego towaru/usługi w innej transakcji. Po drugie, pewność, że dane transakcji zostaną zapisane bez możliwości ich zmiany w znanej weryfikowalnej strukturze. Co więcej, w przypadku większej liczby danych problem kosztu weryfikacji danych historycznych można zmniejszyć, manipulując liczbą transakcji zamykanych w blok.

Koszt, jaki musi ponieść użytkownik takiego systemu, to czas działania podsystemu autoryzacji, który jak zbadano, w ograniczonym środowisku jest znikomy.

## 6. Zakończenie

Osiągniętym rozwiązaniem badawczym jest skuteczna adaptacja koncepcji łańcucha bloków do realizacji zdecentralizowanego podsystemu autoryzacji transakcji modelowanego systemu barterowego.

Uzyskane rezultaty implikują możliwość stosowania rozwiązania jako rozszerzenia dla istniejących systemów transakcyjnych, zapewniającego bezpieczne i bezkonfliktowe przechowywanie danych o transakcjach.

W dalszych pracach autorzy planują badanie znamienych rynkowo własności systemów barterowych zabezpieczonych poprzez *blockchain* w procesie weryfikacji przy wykorzystaniu logik temporalnych opartych na rozgałęzionej strukturze czasu. Kluczowym staje się pytanie, czy uzasadniony jest koszt czasu poniesiony na rozproszenie bazy danych w kontekście bezpieczeństwa transakcji barterowych? Pytanie to nabiera nowego wymiaru w świetle pojawiających się cyberzagrożeń i ewentualnych cyberataków ukierunkowanych na destabilizację wymiany gospodarczej w skali lokalnej, regionalnej i globalnej.

## Literatura

- Accenture, 2018, <https://www.accenture.com/us-en/insight-blockchain-technology-how-banks-building-real-time> (dostęp: 28.02.2018).
- Cakmaz Y.E., Alaca O.F., Durmaz C., Akdal B., Tezel B., Challenger M., Kardas G., 2017, *Engineering a BDI Agent based Semantic e-Barter System*, IEEE Conference Publications, Antalya, Turkey, s. 1072-1077.

- Decentralized eSports Platform FirstBlood Raises \$5.5M in Minutes*, 2016, <https://news.bitcoin.com/esports-firstblood-raises-5-5m/> (dostęp: 24.02.2018).
- Delloite, 2016, *Blockchain: the future of shares trading*, <https://www2.deloitte.com/nl/nl/pages/financial-services/articles/2-blockchain-and-the-future-of-share-trading.html> (dostęp: 31.01.2018).
- Demirkol S., Getir S., Challenger M., Kardas G., *Development of an Agent based E-barter System*, proceedings of the 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), Istanbul, Turkey, 2011, s. 193-198.
- Dickson B., 2016, *Blockchain has the potential to revolutionize the supply chain*, <https://techcrunch.com/2016/11/24/blockchain-has-the-potential-to-revolutionize-the-supply-chain> (dostęp: 22.02.2018).
- Forbes, 2017, <https://www.forbes.com/sites/bernardmarr/2017/11/29/this-is-why-blockchains-will-transform-healthcare/#63ade9e21ebe> (dostęp: 22.02.2018).
- Fortune, 2017, <http://fortune.com/2017/09/12/diamond-blockchain-everledger> (dostęp: 23.02.2018).
- Fortune, 2018, <http://fortune.com/2018/01/20/blockchain-identity-civic-silicon-slopes/> (dostęp: 23.02.2018).
- IBM, 2017, *Tree uses for blockchain in banking*, <https://www.ibm.com/blogs/blockchain/2017/10/three-uses-for-blockchain-in-banking/> (dostęp: 2.02.2018).
- McMullen G., 2017, *Blockchain and Law in 2017: Finally friends or still foes?*, <https://medium.com/ipdb-blog/blockchain-and-law-in-2017-f535cb0e06c4> (dostęp: 22.02.2018).
- Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S., 2016, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*, Princeton University Press, Princeton.
- PwC, 2017, *Blockchain, a catalyst for new approaches in insurance*, <https://www.pwc.com/gx/en/insurance/assets/blockchain-a-catalyst.pdf> (dostęp: 31.01.2018).
- Riva Sanseverino E., Di Silvestre M.L., Gallo P., Zizzo G., Ippolito M., 2017, *The Blockchain in Microgrids for Transacting Energy and Attributing Losses*, IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEEExplore.
- Ubiquity, 2018, <https://www.ubiquity.io/web/index.html> (dostęp: 25.02.2018).