Name: Tobias Binnewies

Hochschule Weserbergland
Studiengang: Wirtschaftsinformatik
Studiengruppe: WI67/21
Dozent: Ralf Hesse

Lösungsorientierte Transferarbeit 2 für Semester 5
(Zeitraum vom 04.12.2023 bis 31.01.2024)

Thema:
Eignungsanalyse von Distributed Ledger Systemen in der Finanz Informatik GmbH & Co. KG

Praxispartner
Finanz Informatik GmbH & Co. KG
Laatzener Straße 5, 30539 Hannover

I Inhaltsverzeichnis

II Abkürzungsverzeichnis

DLS	Distributed Ledger System
DL	Distributed Ledger
ERC	Ethereum Request for Comments
EVM	Ethereum Virtual Machine
ETH	Ether (Währung)
EIP	Ethereum Improvement Proposal

III Abbildungsverzeichnis

1 Einleitung

Einleitung hier

2 Theorie

2.1 Distributed Ledger System

2.1.1 Distributed Ledger

Ein Distributed Ledger ist eine Datenbank, die im Konsens geteilt und über ein Netzwerk synchronisiert wird, das sich über mehrere Standorte, Institutionen oder Länder erstreckt.[1] Es ermöglicht, dass Transaktionen und Aufzeichnungen öffentlich und überprüfbar sind, und da es dezentralisiert ist, gibt es keinen einzelnen Ausfallpunkt. Jeder Teilnehmer im Netzwerk hat Zugang zu den Aufzeichnungen, die über dieses Netzwerk geteilt werden, und kann eine identische Kopie der Daten haben. Änderungen oder Ergänzungen am DL werden nahezu in Echtzeit in allen Kopien widergespiegelt, was die Transparenz und Sicherheit erhöht.

Ein Distributed Ledger System ist ein System, das einen Distributed Ledger verwendet, um Transaktionen zwischen Teilnehmern zu verwalten. Am häufigsten wird die Blockchain-Technologie als DL verwendet.[2]

2.1.2 Blockchain

Eine Blockchain ist eine spezielle Form eines Distributed Ledgers, die aus einer Kette von Blöcken besteht, die jeweils die zu speichernden Daten enthalten.[3] Ein Block besteht aus einem Header und einem Body.[4]

Der Header enthält u. a. den Hash des vorherigen Blocks, einen Zeitstempel und die Nummer des Blocks. Außerdem enthält der Header einen Hash des Bodys. So wird gewährleistet, dass die Blöcke - und so auch die darin beinhalteten Daten - nach ihrem Eintrag nicht verändert werden können, ohne alle nachfolgenden Blöcke abzuändern. (So wird von „Block-Confirmation" gesprochen, wenn eine bestimmte Anzahl von Blöcken nach diesem Block hinzugefügt wurden - je mehr desto sicherer -, da er erst dann als „unveränderlich" gilt.[5])

[1] Vgl. hierzu und im Folgenden LedgerAcademy (2023); Majaski (2023).
[2] Vgl. Tamalio (2023).
[3] Vgl. Greeh, Camilleri (2017), S. 16.
[4] Vgl. hierzu und im Folgenden Singhal, Dhameja, Panda (2500), S. 161 ff.; Ethereum.org (2023a).
[5] Vgl. Singhal, Dhameja, Panda (2500), S. 191.

Der Body enthält die eigentlichen Daten, die gespeichert werden sollen. Im Falle einer Kryptowährung sind dies bspw. die Transaktionen, die in diesem Block gespeichert werden.

2.1.3 Kryptowährung

Ein DLS kann viele Anwendungsmöglichkeiten haben. Am wohl bekanntesten ist die Verwendung als Kryptowährung, wie bspw. Bitcoin.[6] Hierbei werden die Transaktionen zwischen den Teilnehmern des Netzwerks durchgeführt, die in einer Blockchain festgehalten werden. So können Transaktionen Peer-to-Peer durchgeführt werden, also ohne dass eine zentrale Instanz diese überprüfen muss, da die Korrektheit der Transaktionen von allen Teilnehmern geprüft werden.

Bestimmste Chains - wie EVM-Chains - heben dieses Konzept auf eine neue Ebene, indem sie es ermöglichen, Smart Contracts zu erstellen.

2.1.4 Ethereum Virtual Maschine Chain

Eine EVM-Chain ist eine Blockchain-Netzwerk, das die Ethereum Virtual Machine (EVM) verwendet, um Smart Contracts auszuführen.[7] Ethereum selbst ist eine EVM-Chain, die die Kryptowährung Ether (ETH) verwendet. Allerdings gibt es noch weitere Chains, die ebenfalls kompatibel mit der EVM sind, wie bspw. Binance Smart Chain (BSC) oder Polygon (MATIC)[8]. So kann für jede dieser Chains der gleiche Code - sowie Tools für dessen Entwicklung - verwendet werden, um Smart Contracts zu erstellen, die dann auf der jeweiligen Chain ausgeführt werden.

2.1.5 Smart Contracts

Smart Contracts (im Sinne von Ethereum) sind Programme, die auf der Ethereum-Blockchain ausgeführt werden.[9] Sie bestehen aus einer Sammlung von Code (ihre Funktionen) und Daten (ihr Zustand), die - ebenso wie ein Benutzer-Wallet - an einer bestimmten Adresse auf der Ethereum-Blockchain (oder einer anderen EVM-Chain - S. 2.1.4) residieren. Smart Contracts sind eine Art von Ethereum-Konto, was bedeutet, dass sie ein Guthaben haben und Ziel von Transaktionen sein können. Sie werden jedoch nicht von einem Benutzer kontrolliert, sondern sind im Netzwerk bereitgestellt und laufen wie programmiert. Benutzerkonten können dann mit einem Smart Contract interagieren, indem sie Transaktionen einreichen, die eine auf dem Smart Contract definierte Funktion ausführen. Außerdem können Regeln / Bedingungen definiert werden, nach den Code automatisch durchgeführt wird. Standard-

6 Vgl. hierzu und im Folgenden Nakamoto (2008), S. 1.
7 Vgl. hierzu und im Folgenden Bianco (2023).
8 Vgl. Chainlist.org (2023).
9 Vgl. hierzu und im Folgenden Ethereum.org (2023c).

mäßig können Smart Contracts nicht gelöscht werden, und Interaktionen mit ihnen sind irreversibel.

2.1.6 ERC20 Token

Es gibt diverse Standarts für Smart Contracts, die bestimmte Funktionen und Eigenschaften definieren. Einer dieser Standarts ist der ERC20 Token Standard, der die Schnittstellen eines Smart Contracts definiert, der als Token verwendet werden soll.[10] Ein Token kann dabei eine beliebige Repräsentation eines Vermögenswertes sein. In diesem Smart Contract wird die Anzahl der Token gespeichert, die eine bestimme Adresse (also Benutzer-Wallet oder Smart Contract) besitzt. Außerdem werden Funktionen definiert, um u. a. Token von einer Adresse zu einer anderen zu versenden, die Anzahl der Token einer Adresse abzufragen und anderen Adressen die Erlaubnis zu erteilen, Token von der eigenen Adresse zu versenden.[11]

2.2 Use Case Zahlungsverkehr

Ein Use Case dieser Technologie in Bereich einer Bank wäre der Zahlungsverkehr. So wird die Blockchain als Datenbank für die Konten der Kunden verwendet. Eine Blockchain erfüllt automatisch durch ihren Aufbau einige Anforderungen an den Zahlungsverkehr, die in herkömlichen Systemen beachtet und umgesetzt werden müssen. So können Transaktionen - also in diesem Fall Einträge in diese Datenbank - nicht rückgängig, nicht verändert und so nicht manipuliert werden. Es gäbe dennoch die Möglichkeit - je nach konkreter Implementierung - bestimmte Sicherheitsmechanismen einzubauen, um bspw. gegen Geldwäsche oder fehlerhaft Buchungen vorzugehen. Außerdem ist ein AuditLog automatisch vorhanden, da alle Transaktionen in der Blockchain gespeichert werden. Aufbauend darauf können Smart Contracts verwendet werden, um bestimmte Finanzprodukte (z.B. Sparverträge, Kredite, ...) oder auch Multisign (per Multisign Contracts) abzubilden und so zu automatisieren. Um den Kontostand eines Kunden widerzuspiegeln, gäbe es mehrere Möglichkeiten:

- Wert als Layer 1 Währung (z.B. ETH) wechseln
- Wert als bestehende Layer 2 Währung (z.B. Stablecoin) wechseln
- Wert als eigene Layer 2 Währung wechseln

Das Problem bei einer Layer 1 Währung ist, dass diese nicht den Euro darstellt. So müsste der Wert immer in eine andere Währung umgerechnet werden, was zu zusätzlichen Kosten führt. Außerdem ist der Wert einer Layer 1 Währung sehr volatil, was zu Problemen führen kann, wenn der Wert des Kontos nicht mit dem Wert der Layer 1 Währung übereinstimmt. Eine Layer 2 Währung, die den Euro darstellt,

[10] Vgl. hierzu und im Folgenden Ethereum.org (2023b).
[11] Vgl. OpenZeppelin (2023); Ethereum.org (2023b).

wäre ein Stablecoin. Dieser ist an den Euro gekoppelt und hat somit immer den gleichen Wert. Das Problem daran - sowie auch bei der Layer 1 Währung - ist, dass diese einen tatsächlichen Wert haben, und so die Bank dieses Geld nicht für eigene Geschäfte verwenden kann. So währe eine eigene Layer 2 Währung sinnvoll, um den Wert eines Kontos darzustellen, ohne dass dieser einen tatsächlichen Wert hat. So kann die Bank diesen Wert für eigene Geschäfte verwenden, ohne dass der Kunde dadurch einen Verlust erleidet. Außerdem kann so gewährleistet werden, dass nur Kunden der Bank diesen Token verwenden können, da die Bank die einzige ist, die diesen Token ausgibt.

2.3 Öffentliches vs. privates Netzwerk

3 Praxis

4 Eignungsanalyse

4.1 Bewertungskriterien

5 Ausblick

6 Quellenverzeichnis

Bianco, A. (2023):
What are EVM Chains?, https://www.datawallet.com/crypto/evm-chains, Stand: 21.12.2023.

Chainlist.org (2023):
Chainlist - Helping users to connect to EVM powered networks, https://chainlist.org, Stand: 22.12.2023.

Ethereum.org (2023a):
Blocks, https://ethereum.org/en/developers/docs/blocks/, Stand: 03.01.2024.

Ethereum.org (2023b):
ERC-20 Token Standart, https://ethereum.org/en/developers/docs/standards/tokens/erc-20/, Stand: 22.12.2023.

Ethereum.org (2023c):
Introduction to Smart Contracts, https://ethereum.org/en/developers/docs/smart-contracts/, Stand: 21.12.2023.

Greeh, A., A. Camilleri (2017):
Blockchain in Education.

LedgerAcademy (2023):
Distributed Ledger Meaning, https://www.ledger.com/academy/glossary/distributed-ledger, Stand: 21.12.2023.

Majaski, C. (2023):
Distributed Ledgers: Definition, How They're Used, and Potential, https://www.investopedia.com/terms/d/distributed-ledgers.asp, Stand: 21.12.2023.

Nakamoto, S. (2008):
Bitcoin: A Peer-to-Peer Electronic Cash System.

OpenZeppelin (2023):
ERC20.sol, https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol, Stand: 22.12.2023.

Singhal, B., G. Dhameja, P. Panda (2500):
TITLEger, o. A., Springer.

Tamalio, M. (2023):
Wie funktioniert die Distributed-Ledger-Technologie?, https://btv-bank.de/wissen/distributed-ledger-technologie/, Stand: 21.12.2023.

IV Anhangsverzeichnis

V Anhang

Anhang 1 - Interviewleitfaden für Technikverantwortlicher/ Entwickler

Interviewfragen	**Bezugskapitel**
1. Frage 1	Kapitel 1
2. Frage 2	eigene Ergänzung

Anhang 2 - Gesprächsprotokoll 1

Protokoll: Gespräch mit X Y
Teilnehmer: X Y - RolleX
Tobias Binnewies - Dual Student
Thema: ThemaX
Dauer: XY min

Frage: Frage1

Antwort: Antwort1

Frage: Frage2

Antwort: Antwort2

Anhang 1 - Template

Anhang 2 - Template