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

I Inhaltsverzeichnis . . . . . **I**
II Abkürzungsverzeichnis . . . . . **II**
III Abbildungsverzeichnis . . . . . **III**
1 Einleitung **1**
2 Distributed Ledger System **1**
2.1 Distributed Ledger 1
2.2 Blockchain 1
2.3 Kryptowährung 2
2.4 Ethereum Virtual Maschine Chain 2
2.5 Smart Contracts 2
2.6 Nodes 3
2.7 Gas Fee 3
2.8 Konsensalgorithmus 3
3 Use Case Zahlungsverkehr **4**
3.1 Datenspeicherung 5
3.1.1 Layer 1 Währung 5
3.1.2 ERC20 Token 5
3.1.2.1 Stablecoin 5
3.1.2.2 Neuer Token 6
3.2 Anonymität 6
3.2.1 Account-based vs. UTXO-based Chain 6
3.2.2 HD-Wallets 7
3.2.3 UTXO-based Token 7
3.3 Öffentliche vs. private Chain 7
4 Praxis **9**
5 Eignungsanalyse **9**
5.1 Bewertungskriterien 10
6 Ausblick **11**
7 Quellenverzeichnis **12**
IV Anhangsverzeichnis . . . . . **IV**
V Anhang **A1**

II Abkürzungsverzeichnis

Abkürzungen alphabetisch sortieren

DLS	Distributed Ledger System
DL	Distributed Ledger
ERC	Ethereum Request for Comments
EVM	Ethereum Virtual Machine
ETH	Ether (Währung)
EIP	Ethereum Improvement Proposal
PoW	Proof of Work
PoS	Proof of Stake
PoA	Proof of Authority
UTXO	Unspent Transaction Output
HD	Hierarchical Deterministic

III Abbildungsverzeichnis

1 Einleitung

Einleitung:
- Es wird auf den Zahlungsverkehr eingegangen
- Es wird im Scope einer Bank betrachtet, nicht im Scope des gesamten Zahlungsverkehrs / Finanzsystems

Keine Wertung in diesem Kapitel, erst dann in Eignungsanalyse

Genauer erklären

2 Distributed Ledger System

2.1 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]

2.2 Blockchain

evtl. ausführlicher

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

[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 (2018), S. 161 ff.; Ethereum.org (2023a).

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])

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.

2.3 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.

2.4 Ethereum Virtual Maschine Chain

EVM-Chains heben das Konzept DLS auf eine neue Ebene, indem sie es ermöglichen, nicht nur Transaktionen abzuwicklern, sondern dazu noch vorprogramierten Code (Smart Contracts) auszuführen und so viele neue Anwendungsmöglichkeiten erschaffen.[7] Eine EVM-Chain ist also eine Blockchain-Netzwerk, das die Ethereum Virtual Machine (EVM) verwendet, um Smart Contracts auszuführen. 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 können.

2.5 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. 2.4) residieren. Smart Contracts sind eine Art von Ethereum-Konto, was bedeutet, dass sie ein Guthaben haben und Ziel von Transaktionen sein können.

[5] Vgl. Singhal, Dhameja, Panda (2018), S. 191.
[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 (2023d).

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

2.6 Nodes

Teilnehmer des Netzwerks, die über die gesamte Blockchain verfügen und die Transaktionen und Blöcke validieren werden Nodes genannt.[10] Diese Nodes formen das dezentralisierte Netzwerk, das die Blockchain betreibt. Diese sind es auch, die den Code der Smart Contracts und die Transaktionen ausführen. Außerdem validieren sie die Transaktionen und Blöcke, die von anderen Nodes erstellt wurden, und erstellen neue Blöcke, die sie dann an das Netzwerk senden. Dazu verwenden sie einen Konsensalgorithmus, der bestimmt, welche Blöcke gültig sind (s. 2.8). Sowohl für das Ausführen der Smart Contracts als auch für das Validieren der Blöcke erhalten die Nodes eine Belohnung in Form von ETH (oder einer anderen Kryptowährung, je nach Chain), diese Belohnung wird Gas genannt.

2.7 Gas Fee

Gas erklären

2.8 Konsensalgorithmus

Um sicherzustellen, dass bestehende Blöcke nicht verändert werden können und der Inhalt neuer Blöcke valide ist, wird ein Konsensalgorithmus verwendet.[11] Die bekanntesten Algorithmen sind:

- **Proof of Work (PoW):** Miner (Nodes) konkurrieren miteinerander, um ein mathematisches Problem zu lösen.[12] Dem ersten, dem dies gelingt, erhält die Blockbelohnung und kann den nächsten Block erstellen. Um einen Block zu erstellen wird also Zeit benötigt, ein Angreifer müsste also eine höheren Rechenleistung als die Hälfte des Netzwerks besitzen, um die Blockchain zu manipulieren.
- **Proof of Stake (PoS):** Es wird zufällig eine Node ausgewählt, die den nächsten Block erstellen darf und so die Belohnung erhält.[13] Es muss vorab eine Sicherheitsleistung (Stake) hinterlegt werden, die verloren geht, wenn die Node

[10] Vgl. hierzu und im Folgenden Bitcoin.org (2023); Ethereum.org (2023f).
[11] Vgl. Nakamoto (2008), S. 2 f.
[12] Vgl. hierzu und im Folgenden Nakamoto (2008), S. 3.
[13] Vgl. hierzu und im Folgenden McKinsey & Company (2023), S. 2.

versucht, die Blockchain zu manipulieren. Je höher der Stake, desto höher die Wahrscheinlichkeit, dass die Node ausgewählt wird. So wird verhindert, dass ein Angreifer die Blockchain manipuliert, da er mehr als die Hälfte des Stakes besitzen müsste, um die Blockchain zu manipulieren.

- **Proof of Authority (PoA):** Es wird eine Liste von Nodes festgelegt, die die Blockchain validieren dürfen.[14] Dieser Algorithmus wird häufig bei privaten Chains verwendet, da den Nodes vertraut werden muss. So kann die Blockchain nicht manipuliert werden, ohne dass eine der Nodes dies zulässt. Dieser Algorithmus ist sehr schnell, da keine Rechenleistung benötigt wird oder eine Auswahl getroffen werden muss, um einen Block zu erstellen.

3 Use Case Zahlungsverkehr

Zahlungsverkehr:
- Je nach Aufbau kann nur der Kontoinhaber selbst eine Transaktion vornehmen
- Transaktionen können nicht rückgängig gemacht werden –> Rückbuchung durch erneute Transaktion
- Transaktionen können nicht verändert werden –> Keine Manipulation möglich

- AuditLog ist automatisch durch Technologie vorhanden
- Durch Smart Contracts können bestimmte Finanzprodukte (z.B. Sparverträge, Kredite, ...) oder auch Multisign (per Multisign Contracts) abgebildet werden und so automatisiert werden
Großrechnerabschaffung

Probleme:
- Transaktionen sind öffentlich einsehbar –> Keine Privatsphäre (außer evtl. durch HD-Wallets)
- Euro keine Kryptowährung –> Umwandlung in Token (Layer2) notwendig
- Weiterentwicklung bei bspw. neuen Anforderungen schwierig –> Durch Proxys möglich

QUELLEN!?

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 -

[14] Vgl. hierzu und im Folgenden BinanceAcademy (2023).

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.

Ausführlicher & Besipiele / konkrete Idee der Implementierung

Ausführlicher & Besipiele / konkrete Idee der Implementierung

3.1 Datenspeicherung

Um den Kontostand eines Kunden widerzuspiegeln, gibt es folgende Möglichkeiten.

3.1.1 Layer 1 Währung

Bei der Darstellung des Kontostandes als Layer 1 Währung (z. B. ETH) müsste jeder Kunde ein Wallet erhalten / besitzen, dass den Wert des Kontos in einer Layer 1 Währung enthält. Das Problem dabei 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. Die Darstellungsart kommt also nicht in Frage.

3.1.2 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.[15] 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.[16]

3.1.2.1 Stablecoin

Es gibt bestimmte ERC20 Token, die den Wert anderer Assets (unter anderem auch den Euro) abbilden. Diese werden Stablecoin genannt.[17] Das Problem daran - sowie

[15] Vgl. hierzu und im Folgenden Ethereum.org (2023b).
[16] Vgl. OpenZeppelin (2023); Ethereum.org (2023b).
[17] Vgl. hierzu und weiterführend Presidents's Working Group on Financial Markets, Federal Deposit Insurance Corporation, Office of the Comptroller of the Currency (2021), S. 4.

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.

3.1.2.2 Neuer Token

Es wäre also die Erstellung eines eigenen Tokens 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. Außerdem muss der erstellte Token nicht zwingend die genauen Schnittstellen eines ERC20 Tokens, sondern lediglich die Anforderungen der Bank erfüllen.

3.2 Anonymität

In einem öffeltichen Blockchain-Netzwerk sind alle Transaktionen öffentlich einsehbar. Anonymität wird dadurch gewährleistet, dass die Identität eines Teilnehmers von der Walletaddress getrennt ist.[18] Allerdings können mehrere Transaktionen eines Wallets miteinander in Verbindung gebracht werden, und so die Anonymität eines Teilnehmers gefährden. So wird - zumindest bei Bitcoin - dazu geraten jede Address nur genau zweimal zu verwenden. Einmal um Bitcoin zu empfangen und einmal um dieses wieder zu versenden.[19] Dieses Problem würde ebenfalls bestehen, wenn die Kontostände in einem öffentlichen Netzwerk abgebildet werden würden.

3.2.1 Account-based vs. UTXO-based Chain

Unterscheid zwischen Account- und Transactions-basierten Chains erklären –> Account-based Problem: Keine HD-Wallets einfach möglich, so weniger Anonymität

Bei einer UTXO-based Chain (bspw. Bitcoin) werden die Anzahl an Coins pro offener Transaktion gespeichert.[20] Eine Adresse besitzt also genauso viele Coins wie die Summe der offenen Transaktionen, die an diese Adresse gesendet wurden. Eine Transaktion kann mehrere Ein- und Ausgänge haben, wobei die Summe gleich sein muss. So können auch zwei offene Transaktionen unterschiedlicher Adressen zusammengefasst und einem Empfänger zugesendet werden.

Bei einer Account-based Chain (bspw. EVM-Chains) werden die Anzahl an Coins pro Account / Adresse gespeichert.[21] So wird bei einer Transaktion die Anzahl an Coins von einem Account auf einen anderen Account übertragen, sprich die Anzahl an Coins wird von einem Account abgezogen und dem anderen Account hinzugefügt. Die Art von Chain wird vor allem bei Smart Contract fokussierten Blockchains verwendet. Im in diesem Modell Coins mehrerer Adressen zusammenzufassen, sind mehrere Transaktionen notwendig.

[18] Vgl. Nakamoto (2008), S. 6.
[19] Vgl. BitcoinDeveloper (o. J.)
[20] Vgl. hierzu, im Folgenden und weiterführend Singhal, Dhameja, Panda (2018), S. 182 ff.
[21] Vgl. hierzu und im Folgenden Crypto APIs Team (2022).

3.2.2 HD-Wallets

Bei einem HD-Wallet werden beliebig viele Keys (sprich Adressen) aus einem Hauptschlüssel (Seed) generiert.[22] So kann ein Wallet verwendet werden, dabei aber für jede Transaktion eine neue Adresse verwendet werden und so eine hohe Anonymität gewährleistet werden. Diese Art von Wallet sind der heutige Standart für UTXO-based Chains. Allerdings können HD-Wallets nicht für Account-based Chains verwendet werden, da diese keine Keys verwenden, sondern die Anzahl an Coins pro Account speichern.

3.2.3 UTXO-based Token

Idee mit UTXO-based Token auf Account-based Chain einfügen (bzgw. HD-Wallet fähigen Token) –> UTXO vermutlich besser, weil weniger Chain Einträge / Auslesungen notwendig sind

Weiterschreiben

Aufbauend auf der Idee einen neuen Token zu kreieren (s. 3.1.2.2) ...

3.3 Öffentliche vs. private Chain

Öffentliches vs. privates Netzwerk:
PRivates:
- Nur für bestimmte Teilnehmer (in dem Falle dann nur FI)
- wirds wirklich nur von einer Authority verwendet können die Daten geändert werden
- ëigene"Regeln (bspw. kein Gas)

Öffentliches:
- Anonymität schwierig zu gewährleisten
- Transaktionskosten
- Daten können nicht geändert werden

Bei der Verwendung einer Blockchain gibt es die Möglichkeit, einer öffentlichen Chain „beizutreten“ oder dafür eine private Chain zu betreiben.[23] Eine öffentliche Chain ist für alle Teilnehmer offen und kann von jedem verwendet werden.[24] Eine privatee Chain hingegen ist nur für ausgewählte Teilnehmer zugänglich und wird i.d.R. von einer oder mehreren Organisationen / Unternehmen betrieben. Die Auswahl der Art der Chain kann an den Punkten Sicherheit / Unveränderlichkeit, Leistung, Kosten, Berechtigungen und Datenschutz / Anonymität erfolgen:

Folgendes eher in Definition oder späterer Analyse?

- **Sicherheit / Unveränderlichkeit:** Die Sicherheit und Unveränderlichkeit einer Blockchain wird durch ihren Konsensalgorithmus bestimmt. Eine öffentliche Chain wird durch die Interaktion von Tausenden unabhängigen Nodes

[22] Vgl. hierzu und weiterführend Binnewies (2023), S. 8 ff.
[23] Vgl. Ethereum.org (2023e).
[24] Vgl. hierzu und im Folgenden Ethereum.org (2023c).

gesichert, die von Einzelpersonen und Organisationen auf der ganzen Welt betrieben werden. Private Chains haben typischerweise eine kleine Anzahl von Nodes, die von einer oder wenigen Organisationen kontrolliert werden. Diese Nodes können streng kontrolliert werden, aber nur wenige müssen kompromittiert werden, um die Chain umzuschreiben oder betrügerische Transaktionen durchzuführen.

- **Leistung:** Bei privaten Chains können Hochleistungsnodes mit besondererer Hardware sowie ein anderer Konsensalgorithmus verwendet werden, um einen höheren Tansaktionsdurchsatz auf der Basisschicht (Layer 1) erreichen. Bei einer öffentlichen Chain kann ein hoher Durchsatz mit Hilfe von Layer 2 Skalierungslösungen erreicht werden.

- **Kosten:** Die Kosten für den Betrieb einer privaten Chain spiegeln sich hauptsächlich in der Arbeit wider, die Chain einzurichten und zu verwalten, und den Servern zu betreiben, auf denen sie läuft. Während es keine Kosten gibt, um sich mit dem Ethereum Mainnet zu verbinden, muss die Gas Fee (s. 2.7) für jede Transaktion in Ether bezahlt werden. Abhilfe können Transaction Relayers (s. **??**) sein, sodass ein Endkunde diese Gebühr nicht selbst tragen muss.

  Einige Analysen haben gezeigt, dass die Gesamtkosten für den Betrieb einer Anwendung auf einer öffentlichen Chain niedriger sein können als beim Betrieb einer privaten Chain.[25]

Die Quelle genauer erläutern?

- **Berechtigungen:** Bei privaten Chains können nur autorisierte Teilnehmer Nodes einrichten und Transaktionen durchführen. Bei öffentlichen Chains kann jeder Node einrichten und Transaktionen durchführen. So kann ebenfalls jeder auf jeden Contract zugreifen, also dessen gespeicherte Daten auslesen und Funktionen aufrufen. Daher müssen erstellte Contracts so implementiert werden, dass sie nur von den gewünschten Teilnehmern verwendet werden können (s. **??**).

- **Datenschutz / Anonymität:** Der Zugang zu Daten, die auf privaten Chains festgehalten wurden, kann frei vom Betreiber kontrolliert werden. Alle Daten, die auf einer öffentlichen Chain geschrieben wurden, sind für jeden einsehbar, so dass sensible Informationen off-chain gespeichert und übertragen oder verschlüsselt werden sollten. Es bestehen Designpattern, die dies erleichtern, sowie Layer 2 Lösungen, die Daten abgetrennt und von Layer 1 fernhalten können (s. **??**).

  Ebenso sind alle Transaktionen auf einer öffentlichen Chain öffentlich einsehbar, sodass die Anonymität der Teilnehmer nicht gewährleistet werden kann (s. **??**).

[25] Vgl. EYBlockchain (2019), S. 14.

Fazit / Empfehlung welche Chain verwendet werden soll
–> Öffentlich, da sonst nicht dezentral

4 Praxis

Ist-Zustand:
Quelle: Persönliches Gespräch mit Nils Pr... (OE-XXXX)
Portal (Sparkassenmitarbeiter)
Online-Banking (Sparkassenkunde)
Cobol-Job
↓
OSPE-Services
↓
DOING 1 - N
↓
Inhausclearing: DiBus (Quelle: Gespräch Christian Krauthoff (OE-XXXX))
Inland / SEPA: Clearing / TARGET / SWIFT
Ausland: Clearing / SWIFT

Ëigentliche Idee von Cobol wegkommen –> Neues Projekt (AZV) nun aber 80% Cobol"

5 Eignungsanalyse

5.1 Bewertungskriterien

Kriterien:
- Perfomance (Schnelligkeit) –> Durch Layer 2 Lösungen (z.B. Lightning Network) (recht) hoch, ansonsten durch privates System (dann aber nicht dezentral und so Sicherheitsrisiken)
- Skalierbarkeit –> Durch Layer 2 Lösungen (z.B. Lightning Network) möglich
- Sicherheit (u.a. AuditLog) –> Wenn öffentliches System, dann generell durch Implementeirung beeinflusst, ansonsten durch privates System wie jetzt
- Transparenz –> In öffentlichem System sehr hoch, ansonsten durch privates Systeme wie jetzt
- Privatsphäre –> In öffnetlichem System quasi nicht möglich (evtl. durch HD-Wallets), ansonsten durch privates System (dann aber nicht dezentral und so Sicherheitsrisiken)
- Kosten
- Komplexität (Aufwand, Wartung, ...) –> Erstmal Aufwand, dann aber gringer (vor allem wenn nur ein System verwendet wird, im Gegensatz zur aktuellen Situation)
- Erweiterbarkeit (z.B. neue Anforderungen) –> Durch Proxys möglich

Sicherheit:
Ein wichtiger Aspekt, der in der Studie hervorgehoben wird, ist die Blockchain-Sicherheit bzw. Unveränderlichkeit. Die Widerstandsfähigkeit einer Blockchain gegen Manipulationen wird durch ihren Konsensalgorithmus bestimmt. Das Ethereum Mainnet wird durch die Interaktion von Tausenden unabhängigen Knoten gesichert, die von Einzelpersonen und Minern auf der ganzen Welt betrieben werden. Private Ketten haben in der Regel eine kleine Anzahl von Knoten, die von einer oder wenigen Organisationen kontrolliert werden. –> w11

6 Ausblick

Implemetierungsideen:
- HD-Wallets / UTXO-Token
- Proxys
- Transaction Relayers

Weitere Ideen:
- Einbindungen mehrerer Banken –> Privates Netzwerk schaffen (So wie bei Bundesbank beschrieben)

Formulierungen:
... genaueres kann nur eine wirklich praktische Umsetzung / Implementierung zeigen

7 Quellenverzeichnis

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

BinanceAcademy (2023):
Proof of Authority Explained, https://academy.binance.com/en/articles/proof-of-authority-explained, Stand: 08.01.2024.

Binnewies, T. (2023):
Analyse des Aufbaus eines Crypto-Wallets sowie die Einbindung dessen in eine Banking-App am Beispiel der Sparkassen-Banking-App der Finanz Informatik.

Bitcoin.org (2023):
What Is A Full Node?, https://bitcoin.org/en/full-node#what-is-a-full-node, Stand: 08.01.2024.

BitcoinDeveloper (o. J.):
Transactions, https://developer.bitcoin.org/devguide/transactions.html, Stand: 12.01.2024.

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

Crypto APIs Team (2022):
UTXO and Account-Based Blockchains, https://cryptoapis.io/blog/7-utxo-and-account-based-blockchains, Stand: 12.01.2024.

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):
Ethereum Mainnet for Enterprise, https://ethereum.org/en/enterprise/, Stand: 08.01.2024.

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

Ethereum.org (2023e):
Private Ethereum for Enterprise, https://ethereum.org/en/enterprise/private-ethereum/, Stand: 08.01.2024.

Ethereum.org (2023f):
What are nodes and clients?, https://ethereum.org/en/developers/docs/nodes-and-clients/, Stand: 08.01.2024.

EYBlockchain (2019):
Total cost of ownership for blockchain solutions.

Greeh, A. / Camilleri, A. (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
ledgers.asp, Stand: 21.12.2023.

McKinsey & Company (2023):
What is proof of stake?.

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

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

Presidents's Working Group on Financial Markets / Federal Deposit Insurance Corporation / Office of the Comptroller of the Currency (2021):
Report on Stablecoins.

Singhal, B. / Dhameja, G. / Panda, P. S. (2018):
Beginning Blockchain, 1. Aufl., Apress.

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

IV Anhangsverzeichnis

Anhang 1 - Interviewleitfaden für Technikverantwortlicher/ Entwickler A1
Anhang 2 - Gesprächsprotokoll 1 . . . . . A2
Anhang 1 - Template A3
Anhang 2 - Template A3

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