HAW HAMBURG

Rudolf Hecht
Matrikelnummer: [redacted]

Vergleich einer Private Permissioned Blockchain mit traditionellen Datenbanksystemen

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Rudolf Hecht

Vergleich einer Private Permissioned Blockchain mit traditionellen Datenbanksystemen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Klaus-Peter Kossakowski
Zweitgutachter: Prof. Dr. Olaf Zukunft

Eingereicht am: 04 Juli 2022

Rudolf Hecht

Thema der Arbeit

Vergleich einer Private Permissioned Blockchain mit traditionellen Datenbanksystemen

Stichworte

traditionelle Datenbanksysteme, Blockchain, Private Permissioned Blockchain

Kurzzusammenfassung

In dieser Arbeit werden traditionelle Datenbanksysteme in zentrale, verteilte und Clouddatenbanksysteme differenziert und auf konzeptioneller Ebene mit einer Private Permissioned Blockchain (PPB) verglichen. Für diesen Vergleich wurden eigene Kriterien hinsichtlich der Skalierbarkeit, Netzwerksicherheit, Informationssicherheit und Dezentralität aufgestellt. Anhand dieser Kriterien wurden die einzelnen Datenbanksysteme bewertet. Mit den daraus gewonnen Ergebnissen wird die PPB in einer Unternehmensstruktur platziert und erörtert, welche neuen Chancen oder Probleme sich bei einer Enterprise Lösung ergeben. Dies wird an einer praktischen Implikation in einem Medical Health System veranschaulicht.

Rudolf Hecht

Title of Thesis

Comparison of a Private Permissioned Blockchain with Traditional Database Systems

Abstract

In this thesis, traditional database systems are differentiated into centralized, distributed and cloud database systems and compared on a conceptual level with a Private Permissioned Blockchain (PPB). For this comparison, proprietary criteria were established with respect to scalability, network security, information security and decentralization. Based on these criteria, the individual database systems were evaluated. With the results obtained, the PPB is placed in an enterprise structure and discussed what new opportunities or problems arise with an enterprise solution. This is illustrated by a practical implication in a Medical Health System.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

PPB	Private Permissioned Blockchain
IT	Informationstechnik
DLT	Distributed Ledger Technology
P2P	Peer-to-Peer
DBMS	Database Management System
DDBMS	Distributed Database Management System
DDB	Distributed Database
DSGVO	Datenschutz-Grundverordnung
TPS	Transaction per Second
SSD	Solid State drives

RAM	Random Access Memory
MSP	Membership Service Provider
CA	Certificated Authority
DNS	Domain Name Service
DDoS	Distributed Denial-of-Service
LAN	Local Area Network
WAN	Wide Area Network
GAN	Global Area Network
SQL	Structured Query Language
No-SQL	Not only SQL
MDaaS	Malware-Detection-as-a-Service

1 Einleitung

1.1 Motivation

Im aktuellen digitalen Zeitalter gibt es immer noch Unternehmen, die Daten über ein physisches Medium tauschen. Die Gründe dafür sind zum einen Misstrauen gegenüber neuen Technologien, zum anderen die Unwissenheit über die Vorteile dieser. Dieses Phänomen wird im Laufe der Arbeit weiter betrachtet.

Bei der Recherche über die Blockchain-Technologie waren den positiven Möglichkeiten kaum Grenzen gesetzt. Dies führte zur Motivation, mich tiefergehend mit der Thematik zu beschäftigen und welche gewinnbringenden Perspektiven sich durch die Blockchain ergeben.

Bekannt wurde die Blockchain-Technologie insbesondere durch den Finanzsektor (bspw. Bitcoin), weswegen der Großteil des aktuellen Forschungsstandes sich im Bereich Ökonomie bewegt. Eine erweiterte Nutzung dieser Technologie in anderen Sektoren wird nun zunehmend diskutiert. Aufgrund der Einzigartigkeit und des umfangreichen Publikationsstandes zu dieser Technologie wird sich diese Arbeit thematisch insbesondere mit der Public Permissioned Blockchain (PPB) im Vergleich mit traditionellen Datenbanksystemen befassen. Anhand dieses Vergleiches wird geprüft, ob sich abseits der bisherigen Nutzungsfelder ein zusätzlicher Nutzen der Blockchain-Technologie ergibt.

1.2 Zielsetzung

Ziel der vorliegenden Arbeit ist es, zunächst aufzuzeigen, wie eine Blockchain aufgebaut ist. Im Anschluss soll erläutert werden, mit welcher Zugriffsbeschränkung die Blockchain mit anderen traditionellen Datenbanksystemen verglichen werden kann.

Für einen Vergleich der Datenbanksysteme sind ausgewählte Kriterien notwendig. Diese werden in Kapitel 3 wie folgt definiert: Skalierbarkeit, Netzwerksicherheit, Informationssicherheit und Dezentralisierung. Im Anschluss wird die PPB-Technologie und die traditionellen Datenbankensysteme hinsichtlich der Kriterien analysiert.

Mit den gewonnenen Erkenntnissen soll aufgezeigt werden, in welchen Kriterien ein Blockchain-Datenbanksystem besser oder schlechter als ein traditionelles Datenbanksystem funktioniert Durch die gewonnenen Erkenntnisse werden dann Handlungsempfehlungen abgeleitet und vorgestellt, um aufzuzeigen, welche neuen Chancen oder Probleme sich beim Einsetzen eines Blockchain-Datenbanksystems in einem Unternehmen ergeben.

Im Zentrum der Arbeit stehen dabei die folgenden Forschungsfragen:

Forschungsfrage 1: Inwieweit ist die Blockchain-Zugriffsbeschränkung Private Permissioned für ein Unternehmen geeignet?

Forschungsfrage 2: Welche Kriterien lassen sich definieren, um die PPB-Technologie mit traditionellen Datenbanken vergleichen zu können?

Forschungsfrage 3: Wofür ist die PPB hinsichtlich der diskutierten Kriterien besser geeignet als traditionelle Datenbanksysteme?

1.3 Methodik und Struktur

Für das Beantworten der Forschungsfragen wird kein praktischer Versuch in Betracht gezogen, da alle genannten Technologien schon durch wissenschaftliche Vergleiche untersucht wurden. Die Grundlage dieser Arbeit bildet aktuelle, relevante sowie forschungsbasierte Literatur zum vorliegenden Thema. Zunächst werden die technischen Grundlagen einer Blockchain erarbeitet. Für den späteren Vergleich werden die Technologien anhand der Kriterien analysiert.

Daraufhin werden die Argumente in einer Tabelle für jedes Kriterium zusammengefasst. Zuletzt wird eine Diskussion über die Ergebnisse und die genutzten Quellen geführt. Aus den resultierenden Ergebnissen soll eine praktische Handlungsempfehlung folgen.

1.4 Abgrenzung

In dieser Bachelorarbeit werden die hier vorgestellten Datenbanken als vollständige Systeme angenommen, welche verschiedene Konzepte für eine Datenbank implementieren. Allerdings werden nicht die spezifischen Datenbankkonzepte miteinander verglichen, sondern die zusammenhängenden Systeme inklusiver Datenbank. Des Weiteren wird innerhalb einer PPB nicht verglichen welcher Konsensmechanismus am besten geeignet ist. Die Weiterentwicklung der derzeitigen Blockchains und Konsensmechanismen sind vielfältig und können hier nicht umfänglich vorgestellt werden.

Diese Arbeit ist unabhängig von Konsensmechanismen. Lediglich für die praktische Handlungsempfehlung wird ein Konsensmechanismus empfohlen. Zuletzt ist zu beachten, dass die hier vorgestellte PPB in einen konzeptionellen Kontext gesetzt wird, in dem die als Datenbanksystem zu betrachten ist, um die hier vorgestellten traditionellen Datenbanksysteme mit der PPB vergleichen zu können.

1.5 Zielgruppe

Diese Arbeit orientiert sich an Leserinnen und Leser mit einem abgeschlossenen Informatikstudium, die erfahren möchten, inwiefern sich eine PPB in einem Unternehmen einsetzen lässt und welche Probleme damit einhergehen können.

2 Grundlagen für die Technologien

Für die Blockchain-Technologie sind zunächst einige technische Grundlagen notwendig, in denen die Kernelemente erläutert werden. Im Anschluss daran werden die traditionellen Datenbanksysteme vorgestellt.

2.1 Blockchain

2.1.1 Definition

Im Allgemeinen wird die Blockchain folgendermaßen definiert: „Die Blockchain ist eine neuartige Kombination altbewährter Technologien, die spezifische konzeptionelle und technische Merkmale aufweist. Sie besteht aus einer sicheren Datenstruktur, den Blöcken, die in einer kontinuierlichen Liste gespeichert und in einem dezentralen Netz von Nodes verarbeitet werden. Eine digitale Zeichenfolge bildet eindeutige Eigentumsrechte ab. Diese Rechte werden digital übertragen und einem neuen Besitzer zugeordnet.“ [1]

2.1.2 Herkunft

Wie von Rehfeld beschrieben, basiert die Blockchain auf altbewährten Technologien. Die Distributed-Ledger-Technologie (DLT) ist eine davon, zudem auch eine der relevantesten, denn damit ist es möglich, die Transaktionen festzuhalten und auf die einzelnen Teilnehmer zu verteilten. Die Distributed-Ledger-Technologie ist für die eingesetzten Blockchains die Datenbank. Diese soll nun im Detail näher erläutert werden.

Distributed-Ledger-Technologie

Ein Distributed-Ledger (wörtlich „verteiltes Kontobuch“) ist ein öffentliches, dezentral geführtes Kontobuch. Es ist die technologische Grundlage virtueller Währungen und dient dazu, im

[1] (Rehfeld, 2020)

digitalen Zahlungs- und Geschäftsverkehr Transaktionen von Nutzer zu Nutzer aufzuzeichnen, ohne dass es einer zentralen Stelle bedarf, die jede einzelne Transaktion legitimiert.[2]

Die Charakteristiken einer DLT können im Detail noch weiter ausgeführt werden. „Distributed" ist die Technologie durch das verteilte Halten des Registers. Jeder Teilnehmer in dem Netzwerk führt eine eigene Version des Ledger. Durch die Manipulationssicherheit und der Unveränderlichkeit wird sichergestellt, dass die Versionen des Ledger konsistent gehalten werden. Dafür werden alle verifizierten Transaktionen gespeichert und danach automatisch an die Teilnehmer verteilt. Die gespeicherten Hash-Werte sorgen dafür, dass festgestellt werden kann, ob die Daten vollständig, korrekt und frei von Widersprüchen sind. Dies führt zu einer höheren Transparenz dem Netzwerk gegenüber.[3]

Um die Blockchain im Ganzen nachvollziehen zu können, werden noch weitere technische Grundlagen aufgegriffen, wie Peer-to-Peer, Kryptografie anhand von Hash-Funktionen und die Zugriffsbeschränkungen für die Blockchain sowie das Validieren von neuen Blöcken.

2.1.3 Technische Grundlagen

Peer-to-Peer (P2P)

Wurde ein Netzwerk aufgebaut, indem die Teilnehmer ihre Informationen miteinander teilen können, braucht es eine Kommunikationsform. P2P-Netzwerk ist eine Schlüsselkomponente für die Blockchain-Technologie, da P2P die Distributed-Ledger-Technology unterstützt.[4]

„Ein Peer-to-Peer-Netzwerk ist wortwörtlich ein Netzwerk, das gleichrangige Komponenten mit anderen Gleichrangigen direkt verbindet, ohne zentrale Instanzen oder Kontrollen. Es gibt keine Teilnehmer, die die Rolle einer tonangebenden Ordnungsbehörde spielen."[5] Damit ist die Schlüsselkomponente für Public-Blockchains gegeben, da alle teilnehmenden Nodes gleich

[2] vgl. (Geiling, 2016)

[3] vgl. (Lewin, et al., 2019)

[4] vgl. (Learn, 2022)

[5] (Mahlmann & Schindelhauer, 2007)

sind und keine dritte Instanz benötigt wird, die die Kommunikation untereinander und sämtliche Transaktionen kontrollieren. Die folgenden drei Eigenschaften machen das heutige P2P-Netzwerk aus: [6]

1. Client und Serverfunktionalität: In einem P2P-Netzwerk kann jedes Node im Kontext einer Anwendung Daten speichern, senden und empfangen. Es ist damit in der Lage, sowohl Client als auch Serverfunktionalität zu leisten. Im Idealfall sind alle Nodes gleichrangig.

2. Direkter Austausch zwischen Peers[7]: Zwei Nodes eines Netzwerkes können direkt miteinander, in Echtzeit interagieren. Es gibt keine zentrale Instanz, die die Kommunikation verzögert oder filtert. Dabei ist es unerheblich, welche Daten zu welchem Zweck ausgetauscht werden. Beispiele sind einfache Textnachrichten, Multimediadateien oder der Aufruf von Prozeduren.

3. Autonomie: Dem Node eines P2P-Netzwerks kommt dabei vollkommene Autonomie im Sinne der (Selbst-)Kontrolle ihrer eigenen Aktivitäten zu, d. h. sie allein legt fest, wann und in welchem Umfang es anderen seine Ressourcen zur Verfügung stellt. Als Folge dieser Autonomie ist nicht sichergestellt, dass ein Node dem Netz ständig zur Verfügung steht. Das Netzwerk muss also tolerieren, dass die Nodes nicht permanent online sind.

Damit die Transaktionen in kurze Informationsketten passen, werden Hash-Funktionen genutzt.

Hash-Funktionen

Die grundsätzliche Idee hinter einer Hash-Funktion ist: „[…], dass einerseits die Menge des Zielbereichs wesentlich kleiner ist als die potenzielle Eingabemenge und andererseits möglichst keine Kollisionen auftreten, d. h., dass zwei verschiedene Eingabewerte zu unterschiedlichen Zielwerten führen. […] Diese Eigenschaften ermöglichen es unter anderem, mittels des von der Hash-Funktion generierten Zielwertes auf die Unversehrtheit des Eingabewertes zu

[6] vgl. (Michael, 2011), S.17 ff. und (David, 2002) Chapter 1

[7] Bezeichnung für die gleichrangigen Teilnehmer im Netzwerk

schließen.“ [8] Eine Umkehrung dieser Funktion ist mit den derzeitigen Rechenleistungen nicht möglich. [9]

Damit mehrere Daten effizient geprüft werden können, werden sie in eine Datenstruktur gebracht, die „Hash-Bäume“ oder nach dem Erfinder Ralph Merkle auch „Merkle-Bäume“ heißen. [10](Abb. 2.1 Merkle-Baum)

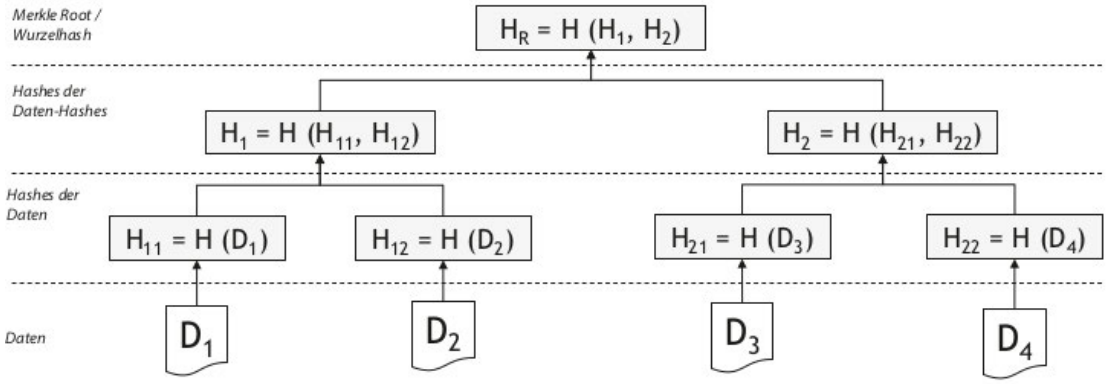

Abbildung 2.1 - Merkle-Baum nach (Merkle, 1987)

Hash-Bäume

Nach dem Buch „Blockchain kompakt – Grundlagen, Anwendungsoption und kritische Bewertung“ von Hans-Georg Fill und Andreas Meier wird die Vorgehensweise der Datenstruktur folgendermaßen wiedergegeben: Jedes der eingefügten Daten wird als Hash mit ihren benachbarten Hash-Werten zusammengefügt und dann wird erneut die Hash-Funktion auf die neuen Daten angewendet. Dieses Prinzip geschieht so lange bis nur noch ein Hash-Wert übrig bleibt, dieser Hash-Wert ist dann der Root-Hash oder Merkle-Root. [11] Die Datenstruktur wird in Abbildung 2.1 nochmal veranschaulicht.

[8] (Meier & Fill, 2020)

[9] vgl. (Frauenhofer-Gesellschaft, 2017)

[10] vgl. (Meier & Fill, 2020)

[11] vgl. (Fill & Meier, 2020)

Der Vorteil dieser Datenstruktur ist, dass jede Veränderung an einem der Dateien den Root-Hash verändert und somit eine schnelle und effiziente Überprüfung erzielt wurde.[12]

Mit den zuvor vorgestellten Methoden und Technologien ist ein Blockchain-Netzwerk anwendbar. Für eine selbstständige und faire Verteilung der Transaktionen muss ein Konsensmechanismus gewählt werden. Konsensmechanismen sorgen für gewünschtes Verhalten im Netzwerk und halten die Teilnehmer des Blockchain-Netzwerks von nicht regelkonformen Handlungen ab, indem sie von ihnen verlangen, einige Ressourcen wie etwa Rechen- oder Speicherkapazität, eine Geldbeteiligung oder Ähnliches bereitzustellen.[13]

Konsensmechanismen

In diesem Abschnitt werden nicht die einzelnen Konsensmechanismen erklärt und vorgestellt, sondern aufgezeigt, welche Ziele diese Mechanismen verfolgen.

In dem Buch „Entwickeln Sie ihre eigene Blockchain – ein praktischer Leitfaden zur Distributed-Ledger-Technologie“ von D. Hellwig, G. Karlic und A. Huchzermeier werden die Kernziele wie folgt beschrieben:[14]

1. Erreichen einer einvernehmlichen Verbindung: Das primäre Ziel jedes Konsensmechanismus ist es, eine einvernehmliche Einigung über den Zustand des Netzwerks zu erzielen. Die Vereinbarung (Konsens) stellt Regeln als Protokoll auf, an die sich strikt von den Netzteilnehmern zu halten ist. Dadurch wird sichergestellt, dass der Netzwerkstatus aktuell gehalten wird.

2. Doppelte Ausgaben verhindern: Das größte Problem von Digitalwährung ist die mehrfache Ausgabe desselben Coin.[15] Ein Konsensmechanismus löst das Problem, indem es nur gültige Transaktionsnachrichten in den öffentlichen Ledger mit aufnimmt, indem jeder Teilnehmer in der Transaktionshistorie prüft, ob der bestimmte Coin daran beteiligt war.

---

[12] vgl. (Fill & Meier, 2020)

[13] vgl. (Hellwig, et al., 2021)

[14] vgl. (Hellwig, et al., 2021)

[15] Physisch ist es erstmal nicht möglich denselben Schein für zwei unterschiedliche Sachen auszugeben.

3. Anreiz zur Selbstregulierung schaffen: Durch Belohnung der Netzteilnehmer und Bestrafung der opportunistischen Akteure, wird ein erwünschtes Verhalten für das Netzwerk erzielt. Die Teilnehmer, die sich an der Konsensbildung beteiligen, werden für ihre Rechenressourcen belohnt (Mining), zweifelhafte Handlungen von Teilnehmern werden finanziell und rechnerisch untragbar gemacht. Somit sorgt der Konsensmechanismus dafür, dass die selbstregulierenden Aspekte eines vertrauensfreien Systems unterstützt werden.

4. Gleichheit gewährleisten: Dadurch, dass die Blockchain ein P2P-Netzwerk ist, wird jeder Teilnehmer zunächst gleichbehandelt und durch das Offenlegen des frei verfügbaren Programmcodes kann bestätigt werden, dass ein Konsensmechanismus allen gegenüber fair handelt.

5. Fehlertoleranz anbieten: Ein Konsensmechanismus sorgt dafür, dass im Falle von Fehlern die Blockchain dennoch verlässlich und konsistent bleibt.

Für ein vollständiges Blockchain-Netzwerk braucht es eine Zugriffsbeschränkung, die regelt, wer an dem Netzwerk teilnehmen darf. Dafür werden öffentliche Blockchains (public), Private-Blockchains (private) und eingeschränkte Blockchains (permissioned) näher erläutert. Mit dem folgenden Abschnitt soll die erste Forschungsfrage erörtert werden: „Inwieweit ist die Blockchain-Zugriffsbeschränkung Private Permissioned für ein Unternehmen geeignet?“

2.1.4 Zugriffsbeschränkungen in einer Blockchain

Für den folgenden Abschnitt werden Begriffe aus dem Buch von Norbert Pohlmann, 2019, „Cyber-Sicherheit“ verwendet.[16]

Für eine Zugriffsbeschränkung wird zunächst zwischen „Permissioned“ und „Permissionless“ unterschieden, hierdurch wird festgelegt wer neue Blöcke in das Netzwerk einfügen darf. Ist ein Netzwerk „Permissioned“ so können nur bestimmte ausgewählte Nodes diese einfügen, wohingegen bei einem Permissionless Netzwerk jeder einen neuen Block einfügen kann.

[16] vgl. (Pohlmann, 2019)

Public Permissionless

Dieses Netzwerkmodell ist das meisterprobte Netzwerk, hierbei kann jeder ohne das Netzwerk beitreten. Dem Node liegt die komplette Blockchain offen und es kann die Transaktionshistorie eingesehen werden. Dem Node ist es auch möglich, neue Blöcke einzufügen, wenn es denn die nötigen Rechenressourcen zur Verfügung stellt und den Wettstreit gewinnt. Hierbei sind die Nodes auch anonym unterwegs und es kann nicht nachgewiesen werden, wer welchen Block hinzugefügt hat. Dieses Modell wird vor allem für die Kryptowährung Bitcoin verwendet.

Public Permissioned

In dieser Blockchain ist der Zugang für jeden möglich, diese wird durch eine Organisation und einem entsprechenden Konsens ausgewählt, wer vertrauenswürdig und berechtigt ist, einen neuen Block hinzuzufügen. Die Wahl für ein Node ist nicht von unendlicher Dauer. Es muss klar definiert sein, warum gerade dieser Node als vertrauenswürdig eingestuft wurde. Dafür wird zum Beispiel das „Practical Byzatine Fault Tolerance (PBFT)"-Verfahren genutzt. Mit diesem Verfahren ist es möglich, viele tausende Transaktionen in der Sekunde zu verarbeiten, wenn es in einer permissioned-Blockchain genutzt wird. Wird ein Node als kompromittiert gemeldet, so kommt die oben genannte Organisation ins Spiel. Diese Gruppe nennt sich „Konsortium". Das Konsortium überprüft das Node und entscheidet, ob der Block dem Netzwerk hinzugefügt werden darf oder nicht. Diese Art von Blockchain wird auch „Consortium Chain" genannt.

Private Permissionless

Der einzige Unterschied zur Public Permissionless Blockchain ist, dass die Nodes sich für die Blockchain registrieren müssen, um Zugriff auf die Blockchain zu erhalten. Somit ist einer zentralen Stelle ersichtlich, wer alles im Netzwerk agiert. Diese Art des Modells ist am wenigsten genutzt.[17]

[17] vgl. (Pohlmann, 2019)

Private Permissioned

Diese Art des Modells ist, die restriktivste Blockchain-Variante. In dieser Blockchain ist der Distributed-Ledger nicht öffentlich und nur ausgewählte Nodes können auf den Ledger schreiben. Es gibt berechtigte Blockchain-Teilnehmer, die neue Blöcke der Blockchain hinzufügen dürfen. Somit ist auch hier eine Registrierung in einer Domäne notwendig,[18] um einen eingeschränkten Zugang zu ermöglichen. Diese lokalisierte Blockchain ist ein Modell, welches für Unternehmen interessant sein könnte, die die Vorteile einer Blockchain nutzen wollen, ohne aber jegliche Transaktionen oder Daten offen legen zu müssen.

Friedrich Thießen beschreibt die Private-Blockchain in seinem Working Paper *Öffentliche vs. Private-Blockchains in der Finanzwirtschaft* aus dem Jahr 2020, folgendermaßen: Eine Blockchain kann aus zwei Blickwinkeln betrachtet werden, zum einen aus der operativen Sicht und zum anderen aus der Strategischen. Mit dem strategischen Blickwinkel können die Abstimmungsregeln eines Unternehmens berücksichtigt werden (Corporate Govermance). Darüber hinaus können komplette Strukturen über Rechte und Pflichten festgelegt werden, um so die operative Ebene mittels Lese- und Schreibrechte kontrollieren zu können. Damit kann genaustens festgelegt werden, ob die Nutzer wie in der Public-Blockchain frei und anonym handeln oder nur verifizierte Nodes agieren dürfen. Somit kann die Blockchain vollständig kontrolliert und administriert werden. Deshalb sind Private-Blockchains aus Corporate-Govermance-Sicht unproblematisch und es kann auf alte bewährte Govermance Strukturen zurückgreifen. Diese Datenbanktechnologie kann in traditionelle Unternehmensstruktur eingebunden werden und bringt ihre eigenen Stärken und Schwächen mit sich.[19]

Da diese Zugriffsbeschränkung die größte Kontrolle über ein Netzwerk gewährt, wird diese Zugriffsbeschränkung für die weitere Ausarbeitung dieser Arbeit empfohlen.

[18] vgl. (Shehri, 2013)

[19] vgl. (Thießen, 2020)

2.1.5 **Smart-Contracts**

Smart Contracts ermöglichen das Automatisieren von Prozessen, Regularien und Organisationsprinzipien. Transaktionen lassen sich um Regeln ergänzen und werden dann zu sog. Smart-Contracts. Sie spezifizieren, was bei einer Transaktion zu prüfen ist und welche Folgeaktivitäten zu initiieren sind. Häufig genannte Beispiele für Smart-Contracts sind elektronische Türschlösser die automatisch prüfen, ob der Nutzer die Nutzungsgebühr bezahlt hat und noch in Besitz der notwendigen Legitimation wie z. B. eines Führerscheins ist. [20]

Neben den vielseitigen und intuitiven Einsatzmöglichkeiten, weisen Smart-Contracts noch ca. 40% Sicherheitslücken auf, die ausgenutzt werden können. Außerdem weisen schon eingesetzte Smart-Contracts auf veränderbare Daten hin oder verweisen auf andere Smart-Contracts, weshalb ein vollständiges Vertrauen in Dritte gemacht werden muss.[21]

Für die technische Umsetzung von Smart-Contracts wird auf die Arbeit des Frauenhofer Instituts verwiesen. „Smart-Contracts machen aus einer Blockchain mehr als nur einen verteilten sicheren Speicher und ermöglichen die automatisierte und vertrauenswürdige Modifikation von Informationen in der Blockchain. So können in Bitcoin Smart-Contracts dazu verwendet werden, verschiedene Arten von Transaktionen wie z. B. Escrow, das heißt die treuhänderische Hinterlegung von Daten zu realisieren. Während Smart-Contracts in Bitcoin nur aus wenigen Operationen bestehen und keine Schleifen realisieren können, bietet die Ethereum-Blockchain eine »quasi-Turing-vollständige« Sprache an, deren Ausführung in einer dedizierten virtuellen Maschine »Gas« kostet. Die Hyperledger-Blockchain geht noch weiter und erlaubt die Ausführung von nahezu beliebigen Programmen. Diese werden „Chaincode“ genannt, der in verschiedenen Hochsprachen wie Java oder Go geschrieben werden kann und von vertrauenswürdigen »Validating Peers« ausgeführt wird. Während der Ausführung hat der „Chaincode“ Zugriff auf die in der Blockchain gespeicherten Informationen und kann sie auslesen oder weitere Informationen speichern. Des Weiteren ist „Chaincode“ bei der Ausführung lediglich durch Docker-Container vom restlichen Environment isoliert, das heißt die Ausführung findet nicht

[20] vgl. (Schütte, et al., 2017)

[21] vgl. (Luu, et al., 2016)

in einer virtuellen Maschine, sondern direkt auf dem Prozessor des Peers statt. Die Korrektheit von Smart-Contracts ist von äußerster Wichtigkeit, da im Gegensatz zu bspw. Desktop- oder Webanwendungen kontinuierliche Updates von Smart-Contracts nicht ohne weiteres möglich sind. Dies bedeutet, dass einmal eingesetzter Smart-Contract-Code nicht mehr ohne Weiteres revidiert werden kann, ohne die Integrität der in der Blockchain gespeicherten Daten in Frage zu stellen.“[22]

2.2 Vorstellung der traditionellen Datenbanksysteme

Grundlegende Informationen

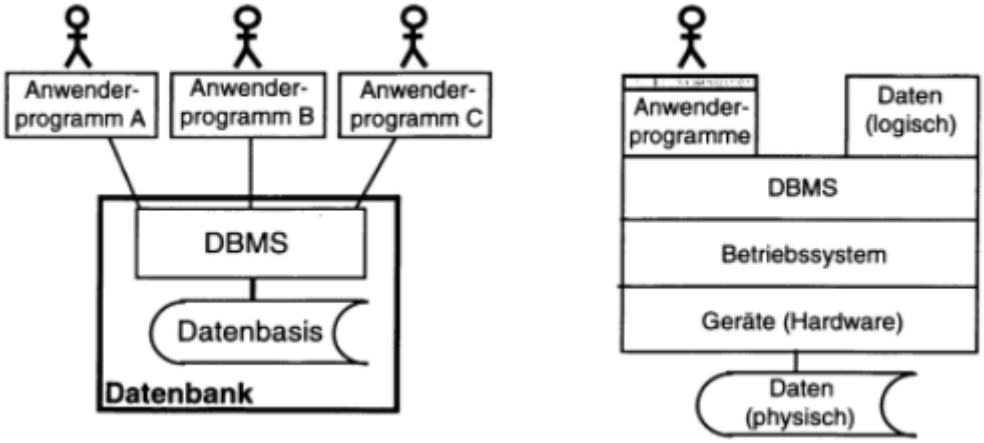

Abbildung 2.2 - Datenbank-Grobdarstellung und -Schichtendarstellung[23]

„Datenbanken und Datenbanksysteme spielen eine wesentliche Rolle in vielen Bereichen der modernen Gesellschaft. Die meisten von uns kommen täglich mit verschiedenen Aktivitäten in Berührung, die die eine oder andere Interaktion mit einem Datenbankmanagementsystem umfassen. […] der Einkauf von Artikeln in einem Supermarkt umfasst heute in den meisten Fällen eine automatische Aktualisierung der Datenbank, die den Bestand der Artikel verwaltet. Die obigen Interaktionen sind Beispiele dessen, was unter **traditionellen**

[22] vgl. (Schütte, et al., 2017)

[23] vgl. (Zehnder, 2013)

Datenbankanwendungen verstanden wird, bei denen die meisten Informationen entweder in Textform oder numerisch gespeichert werden, […]. “ [24]

In der Grobdarstellung einer Datenbank in Abb. 2.2 werden in Datenbasis und DBMS[25] unterschieden. Auf der linken Seite des Abbilds werden Hardwaregeräte zunächst außen vorgelassen, weshalb die Hardwarekomponente eines Servers oder Computers essenziell sind, um auf die physisch liegenden Daten zugreifen zu können. Diesbezüglich schafft die rechte Seite der Abb. 2.2 Abhilfe. Denn neben dem Zugriff auf eine Datenbank durch das DBMS, sind auch Kenntnisse über den Aufbau einer Datenbank von höchster Relevanz. [26]

Relationale Datenbanken bilden die vierte Generation der Datenbanksysteme und sind seit den 80er-Jahren kommerziell verfügbar. Sie sind gekennzeichnet durch eine hohe Unabhängigkeit gegenüber physischen Daten. Sie basieren auf einem einfachen konzeptionellen Modell (Tabellen). Das bedeutet es können Veränderungen sowohl, auf der logischen als auch auf der physischen Seite vorgenommen werden, ohne dass die jeweils andere Seite davon betroffen ist. Die Daten werden für den Benutzer in Form von Tabellen, welche aus einzelnen Zeilen bestehen, dargestellt. Die Kopfzeile enthält die Attribute der Tabelle. Mindestens ein Attribut ist mit einer eindeutigen ID als Schlüssel versehen, um die dazugehörigen Beziehungen der konkreten Werte zuordnen zu können. Somit haben Attribute mit ihren Werten einer Spalte eine Beziehung oder auch eine Relation zueinander. Wird nun eine weitere Tabelle erstellt, mit einer Relation zu Attributen einer anderen Tabelle, sind diese durch diesen Datenpunkt verbunden. [27] Bei den nachfolgenden Datenbanksystemen handelt es sich meist um relationale Datenbanken. Sollte nur von Datenbanken gesprochen werden, so ist eine relationale Datenbank gemeint.

---

[24] vgl. (Elmasri & Navathe, 2002)

[25] Datenbank Management System (DBMS) übernimmt die Aufgabe des Organisierens und Strukturierens von Daten. Des Weiteren ist es die Schnittstelle zum Benutzer, der dann einen lesenden bzw. schreibenden Zugriff erhält.

[26] vgl. (Zehnder, 2013)

[27] vgl. (Vossen, 2008)

Neben den vielen verschiedenen Architekturmöglichkeiten wird sich diese Arbeit auf folgende Modelle konzentrieren: zentrales Datenbanksystem (Client/Server Architektur), Verteilte DBS, Cloud-Datenbanksystem (C-DBS).

2.2.1 Client/Server-Architektur als zentrales Datenbanksystem

Die Architektur beschreibt die Art und Weise, wie Server und deren Clientprozesse miteinander interagieren. Softwarekomponenten sollen so aufgeteilt werden, dass ein Server einen Service und die Ressourcen bereitstellt und ein Client die Benutzerschnittstelle und die Anwendungslogik verwaltet. Diese Komponenten können auf demselben Computer vorhanden sein. Meistens sieht die Architektur ein Netzwerk einer Domäne vor, indem der Server und die Clients räumlich voneinander getrennt sind und sich nur über das lokale Netzwerk austauschen. Letztendlich verhält sich eine Datenbankabfrage eines Clients wie folgt: Der Benutzer erstellt eine Anfrage über Daten und gibt sie in das Anwendungsprogramm ein, die Anwendung prüft die Anfrage und generiert die genutzte Datenbanksprache. Diese wird an den Datenbankserver geschickt, der wiederum die Datenabfrage mit den Serverressourcen verarbeitet und die Ergebnisse an den Benutzer zurücksendet. Wird in einem Netzwerk ein Server, auf dem die Datenbank läuft, genutzt, so wird von einer zentralen Datenbank gesprochen.[28]

Durch diese Architektur kommen folgende Vorteile zusammen:

[28] vgl. (Vossen, 2008)

Ermöglichen voll umfassendem Zugriff auf die existierenden Datenbanken.
Die Arbeitsgeschwindigkeit erhöht sich, da die Kommunikationsprozesse auf mehreren CPUs gleichzeitig ablaufen kann.
Die Hardwarekosten können reduziert werden, da nur der Server für die intensiven Rechen - und Speicherkapazitäten verantwortlich ist.
Netzwerkkommunikation kann gesenkt werden, weil ein Teil des Prozesses vom Client verarbeitet werden kann und nur der direkte Zugriff auf die Datenbank durch das Netzwerk erfolgt.
Die Integritätsprüfung wird vom Server durchgeführt und damit sind die Bedingungen zur Transparenz nur an einer Stelle festgelegt.
Leicht erweiterbar, es kann zu einer Architektur mit verteilten Datenbanken weiterentwickelt werden

Tabelle 2.1 - Vorteile der Client/Server-Architektur [29]

Ist eine Datenbank an mehreren Standorten vorhanden, werden die Informationen auf die Datenbanken verteilt, so wird von einem verteilten Datenbanksystem gesprochen. Dies wird im nächsten Abschnitt im Detail vorgestellt.

2.2.2 **Verteiltes Datenbanksystem**

Verteilte Datenbanken oder Distributed Databases (DDB) werden als folgendes Konzept beschrieben: Ein verteiltes Rechnersystem besteht aus einer Reihe von - nicht unbedingt homogenen - Verarbeitungselementen, die über ein Computernetzwerk miteinander verbunden sind und bei der Durchführung bestimmter Aufgaben zusammenarbeiten. Als allgemeines Ziel sollen große unhandliche Probleme in kleinere aufgeteilt werden (Divide and Conquere), um diese dann koordiniert und effizient zu lösen.

[29] vgl. (Connolly, et al., 2002)

Ein verteiltes Datenbanksystem kann als eine Sammlung mehrerer logisch zusammenhängender Datenbanken, die über ein Computernetzwerk verteilt sind, verstanden werden. Ein verteiltes Datenbankmanagementsystem (DDBMS) ist ein Softwaresystem, das ein verteiltes Datenbanksystem verwaltet und die transparente Verteilung für den Benutzer übernimmt.[30]

In der folgenden Tabelle werden die Vorteile von verteilten Datenbanksystemen aufgezeigt.

Das Verwalten von Daten, die auf unterschiedlichen Transparenzebenen agieren: - Verteilung und Netzwerktransparenz: Dem Nutzer ist nicht bekannt, wo sich die physischen Daten genau befinden. - Replikationstransparenz: Kopien von Daten können an mehreren Orten gespeichert werden, um bessere Verfügbarkeit, Zuverlässigkeit und Performanz zu erzielen. - Fragmentierungstransparenz, eine Anfrage wird evtl. in Fragment Anfragen aufgeteilt, der Benutzer bekommt davon aber nichts mit
Höhere Zuverlässigkeit und Verfügbarkeit: Durch die Verteilung der Daten und DBMS auf mehreren Nodes. Sollte ein Nodes ausfallen, so ist der Service dennoch sichergestellt und die Daten sind zu einem bestimmten Zeitpunkt zuverlässig vorhanden und in einem Zeitintervall kontinuierlich verfügbar.
Bessere Performance: Durch das Verteilen der DBMS wird die Datenbank selbst fragmentiert. Aus einer großen zentralen monolithischen Datenbank werden kleinere, die auf verschiedenen Nodes liegen. Lokale Anfragen können schneller verarbeitet werden, da die Transaktionen pro Nodes geringer wird. Es muss nicht mehr ein Server alle Transaktionen verarbeiten. Mit diesem Vorgehen kann der Anfragenparallelismus erreicht werden.
Leichte Erweiterung beim Hinzufügen von zusätzlichen Benutzern oder Nodes sowie Datenbanken.

Tabelle 2.2 - Vorteile von verteilten Datenbanken[31]

[30] vgl. (Elmasri & Navathe, 2002), S. 822.

[31] vgl. (Elmasri & Navathe, 2002), S. 823f.

Um die Vorteile einer verteilten Datenbank nutzen zu können, obliegt dem DDBMS eine höhere Komplexität. Zusätzlich zu den Aufgaben von zentralisierten Datenbanken und dem Verarbeiten von Anfragen, muss das DDBMS entscheiden auf welchen Nodes, welche Fragmente abgelegt werden, ohne die physische Relation der Daten zu zerstören. Damit Benutzer das Gefühl bekommen, sie arbeiteten an einem zentralen DBMS, muss das DDBMS einen Homogenitätsgrad erreichen, arbeiten alle Clients mit den identischen Versionen. So sind sie als homogen zu bezeichnen - andernfalls sind die Systeme heterogen. [32]

2.2.3 **Cloud Systeme**

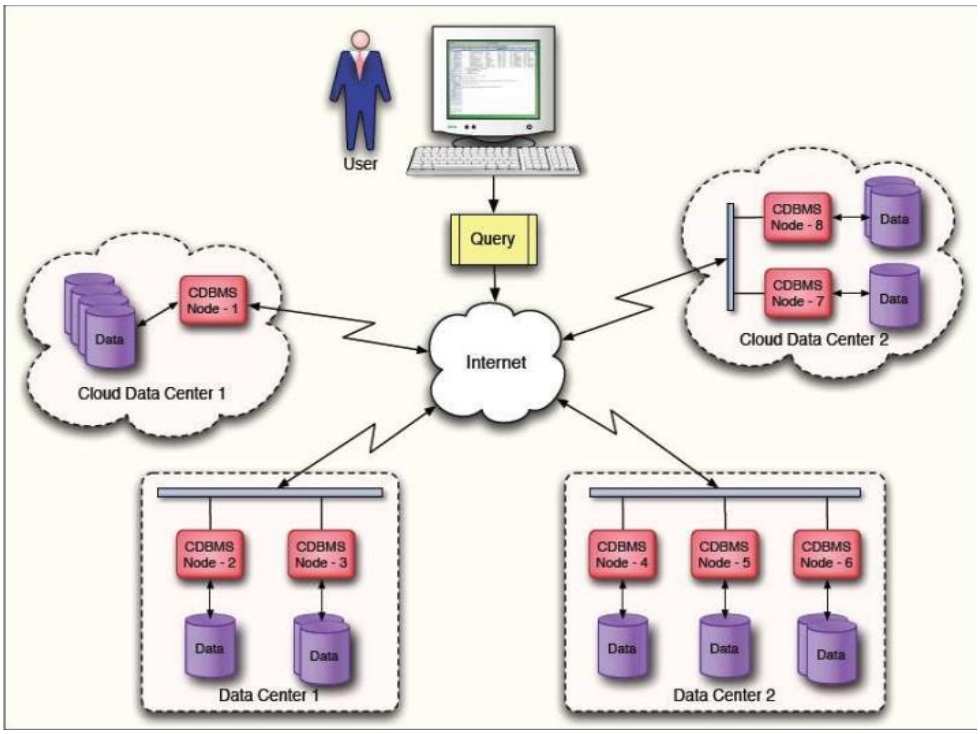

Abbildung 2.3 - Struktur einer Cloud Datenbank [33]

Ein Cloud-basiertes Datenbanksystem unterscheidet sich von anderen Datenbanksystemen in erster Linie darin, dass einem die Hardware eines solchen Servers nicht selbst gehört, sondern

[32] vgl. (Elmasri & Navathe, 2002)

[33] (Shehri, 2013)

dem Cloud-Provider. Ein Cloud-Datenbankzentrum ist zunächst so aufgebaut, dass das Zentrum an einem Ort ist, an dem viele tausende Hardwarekomponenten miteinander verbunden sind. Diese Ressourcen können über eine Applikation wie z.B. vSphere in virtuelle Server verteilt werden, auf diesen virtuellen Server können dann die unterschiedlichsten Applikationen installiert werden, unter anderem eine Datenbank. Dies hat den Vorteil, dass nur so viel Ressource und Verwaltung gezahlt wird, wie auch verbraucht. Sollten für gewisse Szenarien mehr Speicher benötigt werden, so kann man für einen gewissen Zeitraum, Ressourcen aufstocken und bei Beendigung des Szenarios die Ressourcen wieder abgeben. Dies wird als Database-as-a-service verkauft. Um eine hohe Verfügbarkeit auf der Welt zu ermöglichen, gibt es diverse Standorte dieser Serverfarmen. Damit kann eine cloudbasierte Datenbankarchitektur zentrale Komponenten sowie verteilte Komponenten kombinieren. Das Datenbank Managementsystem für eine Cloud-Datenbank ist meist eine Schnittstelle über das Internet. Dieses DBMS sammelt und sortiert die Query's, sodass die am nächsten liegenden Datenzentren ausgewählt werden, die diese Query am schnellsten verarbeiten kann. Hat die Datenbank die Query verarbeitet, erhält der Benutzer die Rückmeldung auf direktem Weg, ohne dass die Antwort über Umwege verzögert wird. [34]

Ein Problem einer Cloud-Datenbank sind Sicherheitslücken und das Einhalten der Privatsphäre von Nutzerdaten. Dadurch, dass die Hardware und die entsprechenden Implementierungen von einer Firma zur Verfügung gestellt werden, können Unternehmen, die diesen Service in Anspruch nehmen, keine zusätzlichen Maßnahmen unternehmen, um die Daten vor neuen Hack-Angriffen zu schützen.[35]

Neben dem grundsätzlichen Problem der Privatsphäre von Clouddaten, hat die EU auch damit zu kämpfen, dass die Daten nicht ohne weiteres in die Datenzentren der USA übermittelt werden. Mit der neuen DSGVO soll ein Standort eines Datenzentrums in der EU dafür sorgen, dass die vor Ort geltenden Datenschutzrichtlinien umgesetzt werden. Das Umsetzen solcher

[34] vgl. aus dem Englischen (Shehri, 2013)

[35] vgl. (Shehri, 2013)

Richtlinien ist schwierig, wenn die Experten der Datenzentren ihren Sitz nicht in der EU haben und bei Störungen oder Problemen aushelfen.[36]

Vorteile von Cloud-Datenbanksysteme sind:

Hohe Flexibilität in der Skalierbarkeit von Hardware
Hohe Verfügbarkeit durch mehrere Datenzentren und verschiedenen Access Points aus dem Netz.
Massive Kostenersparnisse durch das Nichtverwalten eines eigenen Rechenzentrums
Flexibler Zugang zum Service, solange Internetzugriff vorhanden ist.

Tabelle 2.3 - Vorteile eines cloudbasierten Datenbanksystem nach (Pizette & Cabot, 2012)

[36] vgl. (Schonschek & Witmer-Goßner, 2020)

3 Einordnung der Vergleichskriterien

Für das folgende Kapitel werden Kriterien definiert, mithilfe derer die hier vorgestellten Technologien verglichen werden sollen. Der Vergleich der Technologien soll anhand folgender Kriterien erfolgen: Skalierbarkeit, Netzwerksicherheit, Dezentralität und Informationssicherheit. Weshalb diese Kriterien ausgewählt wurden, wird im folgenden Abschnitt erklärt.

3.1 Methodik für den Vergleich

Die oben aufgeführten Kriterien wurden ausgewählt, um ein breiteres Spektrum der Technologien zu erhalten, mit denen zum Schluss eine praktische Handlungsempfehlung getätigt werden kann. Die Erfüllung der nachfolgenden Kriterien sorgt für eine positive Bewertung, wohingegen eine Nicht-Erfüllung zu einer schlechteren Bewertung führt. Für die optische Veranschaulichung der Ergebnisse werden Tabellen verwendet. Die Bewertung der Punkte für die Tabellen sind wie folgt definiert:

Stufe	0	1	2	3	4
Wertung	nicht vorhanden	gering	mittel	gut	vollständig
Stufe in %	0	25	50	75	100

Tabelle 2.4 - Bewertungsskala für die Vergleichskriterien

3.2 Auswahl der Kriterien

Die hier zu diskutierenden Technologien: PPB, zentrales Datenbanksystem, verteiltes Datenbanksystem und Cloud-Datenbanksystem sollen zunächst in einem wirtschaftlichen Unternehmen eingeordnet werden. Um diese Technologien aussagekräftig zu vergleichen, wird sich zunächst an einem der Designziele für ein verteiltes System von Tanenbaum bedient. Skalierbarkeit ist eine Eigenschaft, die eine nachhaltige Technologie auszeichnet, denn damit ist diese Technologie für verschiedene Unternehmensgrößen einsetzbar. Security ist der Überbegriff aus

dem Englischen für Sicherheit, allerdings ist der Begriff Sicherheit breit gefächert. Daher wird dieses Kriterium in zwei Kriterien spezifiziert. Zum einen in Netzwerksicherheit, damit soll festgestellt werden, welche Technologie welche Sicherheitsmaßnahmen einsetzen, um sich vor Angriffen zu schützen. Zum anderen in Informationssicherheit, dabei sollen die zu verwendeten Methodiken der Technologien dargestellt werden, um sicherzustellen, dass die Informationen stets verfügbar und vertraulich sind sowie ihre Integrität aufweisen. Zuletzt soll die Dezentralität aufgegriffen werden, diese Eigenschaft stammt aus dem Blockchain-Trilemma und ist für Public-Blockchains ein wichtiger Faktor.

3.3 Blockchain Trilemma

Dieses Problem in der Blockchain-Technologie wurde von dem Erfinder von Ethereum, Vitalik Buterin, beschrieben. Er fasste das Problem unter dem bis dahin genutzten Begriff „Blockchain Scalability Trilemma“[37] zusammen.

Vorsichtshalber wird erwähnt, dass es sich nicht um das Blockchain-Trilemma für die Konsensmechanismen handelt. Diese Art des Trilemma behandelt das Zusammenspiel von Fehler-Toleranzen, Ressourcen-Effizienz und volle Übertragbarkeit. [38]

[37] vgl. (Buterin, 2020)

[38] vgl. (Abadi & Brunnermeier, 2022)

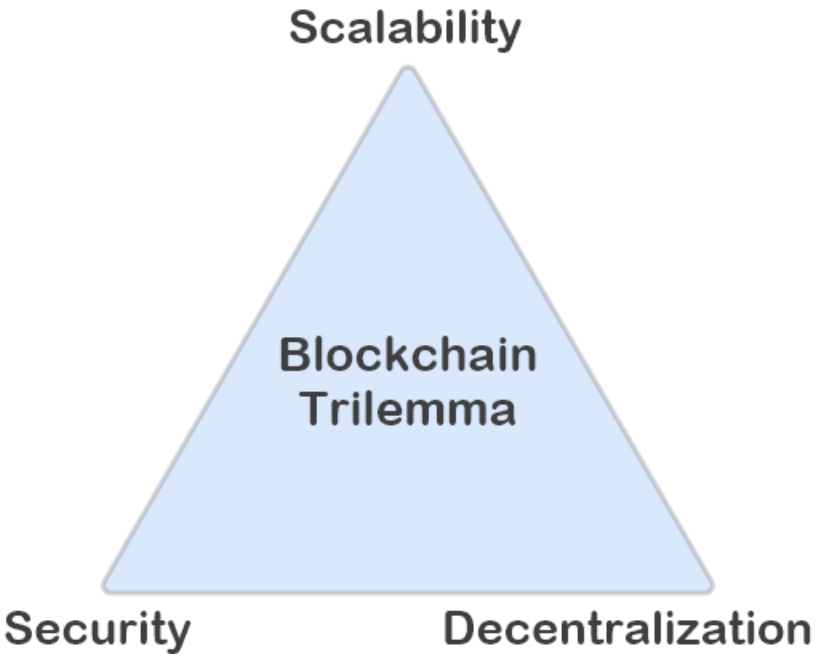

Abbildung 2.4 - Blockchain Scalability Trilemma[39]

Die Skalierbarkeit einer Blockchain ist oft der Schlüssel zum Erfolg. Damit die Blockchain auch für Unternehmen oder große Communitys bestmöglich nutzbar ist, muss die Blockchain viele Nodes und/oder große Datenmengen verarbeiten können. Dies zeichnet sich vor allem darin aus, dass die Skalierbarkeit für Blockchains in Transactions per Second (TPS) bemessen wird.

Für eine Blockchain mit einer hohen Security verspricht sich eine Blockchain ein hohes Vertrauen der Nutzer. Anders als bei den Banken sind alle Transaktionen der Blockchain öffentlich einsehbar. Die Transaktionen in einer Blockchain selbst können nicht gelöscht oder rückgängig gemacht werden. [40] Dies besagt auch die Kerneigenschaft einer Blockchain: Unveränderlichkeit. [41]

[39] (Karaarslan & Konacaklı, 2020)

[40] vgl. aus dem Englischen (Demeester, 2017)

[41] vgl. (Fill & Meier, 2020)

Die Blockchain bringt durch das Nutzen von verteilten Systemen sowie P2P eine gute Dezentralisierung mit sich. Blockchains wollen das Netzwerk selbstständig und automatisch entscheiden lassen, ohne dass eine kontrollierende Instanz Entscheidungen übernimmt. Für ein faires und deterministisches Verhalten werden Konsensmechanismen eingesetzt, die diese Entscheidung von außen ersetzen soll. Damit wäre eine Blockchain in der Lage, Probleme komplett dezentral zu lösen, ohne dass eine zentrale Instanz die Entscheidungen übernimmt.

Für die Public-Blockchains sind die Designentscheidungen so getroffen, dass einerseits auf die Dezentralität nicht verzichtet werden muss. Andererseits kann eine Public-Blockchain derzeit nur zwei der Aspekte des Blockchain Trilemma vollkommen berücksichtigen.[42] Das bedeutet, wenn eine Blockchain sich entscheidet eine höhere Skalierbarkeit (z.B. TPS) zu erreichen, die Dezentralität aber nicht aufgeben möchte, kann die Security nicht mitskalieren.[43] Es gibt Forschungen, wie die Blockchain eine höhere Skalierbarkeit erreichen kann und die Sicherheit mitskaliert. [44]

Im folgenden Abschnitt wird nochmal im Detail erklärt, wie die hier definierten Kriterien zu deuten sind.

3.4 Skalierbarkeit

Skalierbare Technologien sollen in der Lage sein, auf ein dynamisches Wachstum eines Unternehmens reagieren zu können, ohne dabei die bestehende Infrastruktur zu stören. Die skalierbaren Lösungen weisen darauf hin, für welche Unternehmensgrößen sie geeignet sind. Für die traditionellen Datenbanksysteme gelten dieselben Zugangseinschränkungen wie für eine PPB (siehe Kapitel 2.1.4), der Zugriff auf das interne Netzwerk ist nur ausgewählten Nutzern gestattet, diese werden durch zentrale Punkte erteilt und verwaltet. Die Transaktionsgeschwindigkeit (TPS) wird für einen direkten Vergleich der Skalierbarkeit außer Acht gelassen, jedoch als Argument für eine Größen-Skalierbarkeit akzeptiert.

---

[42] vgl. (Barkai, 2001)

[43] vgl. (Buterin, 2020)

[44] vgl. (Songze Li, 2020)

Für das weitere Vorgehen wird die Skalierbarkeit in drei Dimensionen unterteilt, die nach Tanenbaum folgendermaßen beschrieben werden: Größen-Skalierbarkeit, Geographische Skalierbarkeit. Administrative Skalierbarkeit.

Größen-Skalierbarkeit

Für die Größen-Skalierbarkeit wird darauf geachtet, wie die Ressourcen erweiterbar sind. Hierbei geht es um physische Ressourcen sowie die eingesetzte Architektur und Methoden. Außerdem wird aufgezeigt, wie vielen Nutzern Zugriff auf das System gewährt werden kann sowie die einfache Erweiterbarkeit des Systems.

Geographische Skalierbarkeit

Die geographische Skalierbarkeit bedeutet, wie zuverlässig ist eine Verbindung, wenn die Kommunikationspartner viele Kilometer voneinander entfernt sind. Etwas ist gut Geographisch Skalierbar, wenn auch bei vielen Kilometern eine Kommunikation verzögerungsfrei abläuft. Außerdem soll festgestellt werden, wie zuverlässig die genutzten Ressourcen sind.

Administrative Skalierbarkeit

Dabei wird der Verwaltungsaufwand des Systems bemessen. Bei mehreren Administratoren soll bei der Findung geeigneter Richtlinien folgendes berücksichtigt werden: Die Netzwerksicherheit ist so zu schützen, dass niemand unberechtigten Zugang zum Netzwerk erhält, jedoch sollen Nutzer des Netzwerks nur so weit beschränkt werden, wie es notwendig ist.[45]

3.5 Netzwerksicherheit

Unter Netzwerksicherheit kann zunächst viel verstanden werden, daher werden die Worte von Andrew Tanenbaum zitiert: „[…] In its simplest form,
it is concerned with making sure that nosy people cannot read, or worse yet, secretly modify messages intended for other recipients. It is concerned with people trying to access remote services that they are not authorized to use. It also deals with ways to tell whether that message

[45] vgl. (Tanenbaum & Steen, 2018)

purportedly from the IRS ‘‘Pay by Friday, or else’’ is really from the IRS and not from the Mafia […] “.[46]

Damit soll verdeutlicht werden, dass das Kriterium für die Netzwerksicherheit so bewertet wird, wie ein Datenbanksystem sich gegen die gängigsten Angriffe verteidigen kann. Sind die Möglichkeiten gegeben, einen Angriff zu verteidigen, so trägt dies zu einer positiven Bewertung bei.

3.6 Informationssicherheit der Daten

Für die Informationssicherheit der Daten lassen sich nach A. Pfitzmann und M. Rost[47] 3 elementare Schutzziele definieren:

- Verfügbarkeit: Gesicherter Zugriff auf Informationen innerhalb einer festgelegten Zeit
- Vertraulichkeit: Gesicherter Nichtzugriff auf Informationen
- Integrität: Information ist gesichert echt

Verfügbarkeit

„Die Verfügbarkeit ist definiert als die Eigenschaft, dass ein System unmittelbar zur Nutzung steht.“[48] Um dem gerecht zu werden, ist die Verfügbarkeit eng mit der Redundanz[49] verknüpft. Das ermöglicht einen Service nicht nur schneller werden zu lassen, sondern bleibt auch verfügbar, sollte ein Service ausfallen. Damit gewinnt leider nicht nur die hohe Verfügbarkeit, sondern auch der Energiebetreiber, da mit redundanten Systemen auch viele Ressourcen

[46] (Tanenbaum & Wetherhall, 2011 5th ed.)

[47] vgl. (Rost & Pfitzmann, 2009)

[48] vgl. aus dem Englischen (Tanenbaum & Steen, 2018)

[49] Wenn Informationen, Daten oder auch Systemkomponente mehrfach vorhanden sind

eingesetzt werden müssen.[50] Zusätzlich zu vielen redundanten Systemen muss das Problem für inkonsistente Daten im Blick behalten werden.[51]

Vertraulichkeit

Nach Faber E. & Behnsen W. in dem Buch „Join Security Management: organisationsübergreifend handeln", wird Vertraulichkeit wie folgt zitiert „Die Vertraulichkeit von Informationen drückt die Notwendigkeit aus, diese Informationen vor Zugriff durch oder Offenlegung gegenüber Unberechtigten (Personen oder Systemen) zu schützen. Die Vertraulichkeit wird z. B. durch die Einschränkung von Zugriff, Lesbarkeit, Informationsfluss und Auffindbarkeit aufrechterhalten. Beispiele für zugehörige Sicherheitsmaßnahmen sind Rechteprüfung bzw. Zugriffsmanagement (Zugriff), Verschlüsselung (bzgl. Einschränkung der Lesbarkeit), Enterprise Digital Rights Management (bzgl. Informationsfluss) und Steganographie (bzgl. Auffindbarkeit)"[52].

Integrität

„Unter dem Nachweis der Integrität elektronischer Daten versteht man den Nachweis, dass diese vollständig und unverändert sind."[53] So erklärt es das Bundesamt für Sicherheit in der Informationstechnik. Eine gemeinsame Definition des Begriffs gibt es nicht, die Übersetzung des Begriffes, integritas aus dem Lateinischen bedeutet ‚Unversehrtheit' oder ‚Reinheit'. Neben Vertraulichkeit und Verfügbarkeit bildet Integrität das letzte der drei elementaren Ziele der Informationssicherheit.[54]

Dem Nutzer stehen verschiedene Möglichkeiten zur Verfügung, um eine möglichst hohe Datenintegrität zu gewährleisen. Zunächst kann überprüft werden, ob die eigenen Daten vollständig sind. Dafür gibt es die sogenannte Prüfsumme, die berechnet wird, sobald der Datensatz

---

[50] vgl. (Tanenbaum & Wetherhall, 2011 5th ed.)

[51] vgl. (Tanenbaum & Steen, 2018)

[52] (Faber & Behnsen, 2018)

[53] (Anon., 2022)

[54] vgl. (Rost & Pfitzmann, 2009)

vollständig ist, und wird den Nutzern zur Verfügung gestellt. Wird der Datensatz heruntergeladen oder verteilt, können die Empfänger die Prüfsumme selbst berechnen. Sollte die Prüfsumme von der Quelle abweichen, bedeutet dies, dass auf dem Weg zum Ziel dem Datensatz etwas hinzugefügt wurde oder etwas verloren gegangen ist.

Eine andere Methode wäre die Verwendung von digitalen Signaturen. Eine digitale Signatur beschreibt den Prozess sich als Herausgeber von Daten validieren zu lassen. So erhält jede Nachricht eine validierte Signatur, die mit einem Public Key hinterlegt ist. Jede Veränderung an der signierten Nachricht lässt die Signatur selbst ungültig werden und der Nachricht sollte kein Vertrauen mehr geschenkt werden. [55]

3.7 Dezentralisierung

Im folgenden Abschnitt soll erläutert werden, was unter einer vollständigen Dezentralität zu verstehen ist und mit welchen Mitteln eine Dezentralisierung erreicht werden kann.

Definition

Allgemein ist damit die Aufteilung von Verantwortungen und Zuständigkeiten auf mehrere Stellen gemeint. Im Wirtschaftssektor bedeutet der Begriff „Dezentralisation“, dass nicht eine Stelle oder eine Person allein die Entwicklung eines Unternehmens bestimmt.[56] So wird deutlich, wie sich dezentrale von verteilten Systemen unterscheiden. Nach A.S. Tanenbaum ist ein Verteiltes System eine Versammlung von unabhängigen Elementen, die der User als ein einziges kohärentes System wahrnimmt. Äußerlich wirkt es so als hätte ein verteiltes System Eigenschaften eines dezentralen Systems, bei näherer Betrachtung sind es im Backend zentrale Komponenten, die das verteilte System aufrechterhalten. [57]

[55] vgl. (Barbara, 2009)

[56] vgl. (Toyka-Seid & Schneider, 2022)

[57] vgl. (Tanenbaum & Steen, 2018)

Mittel zum Zweck

Das Hauptwerkzeug für eine Dezentralisierung hinsichtlich der Kommunikation zwischen Nodes, ist ein P2P-Netzwerk. Hierbei sind die Nodes, Server und Client zugleich. Dadurch besitzen die Nodes die Fähigkeiten, mit jedem im Netzwerk zu kommunizieren.

Neben der möglichen Kommunikation unter den Nodes braucht es für eine Dezentralisierung die Freiheit durch eine kontrollierende dritte Instanz. Dafür gibt es ein gutes Beispiel in dem Konzept der „Self Sovereign Identity für Deutschland" vom Bundesministerium für Wirtschaft und Klimaschutz. Hierbei sollen die Endnutzer ein System verwenden können, in dem sie selbst Identitätsinformationen speichern und verwalten. Diese Informationen könnten auf ein mobiles Gerät abgelegt werden, um sich bei anderen Institutionen zu identifizieren.[58]

Weitere Möglichkeiten um dezentral zu werden ist eine selbstständige dezentrale Koordination für Prozesse. Dafür wird sich an dem Beispiel aus dem Buch „Distributed Systems" von (Tanenbaum & Steen, 2018) bedient, welches selbst (S.-D., et al., 2004) zitiert. Dieses Beispiel zeigt aber auch ein Problem bei einer dezentralen Koordination: Für eine vollständige dezentrale Lösung wird ein Voting Algorithmus verwendet. Es wird angenommen, dass jede Ressource N-mal repliziert ist. Jedes Replikat hat bei konkurrierenden Prozessen seinen eigenen Koordinator, der den Zugriff kontrolliert. Wenn ein Prozess Zugriff auf die Ressource möchte, benötigt dieser die mehrheitlichen Stimmen von m >N/2 Koordinatoren. Es wird angenommen, dass ein Koordinator einen Prozess darüber informiert, sollte der Zugriff nicht gewährt werden. Es wird davon ausgegangen, dass sich ein abgestürzter Koordinator schnell erholt, aber alle Stimmen vergisst, die er vor dem Absturz abgegeben hat. Eine andere Möglichkeit ist, dass sich ein Koordinator zu beliebigen Zeitpunkten selbst zurücksetzt. Das Risiko besteht darin, dass der Koordinator beim Zurücksetzen vergisst, dass er einem Prozess die Erlaubnis zum Zugriff auf die Ressource erteilt hat. Infolgedessen kann er diese Erlaubnis nach seiner Wiederherstellung fälschlicherweise wieder einem anderen Prozess vergeben.

[58] vgl. (BMWK & Michael, 2022)

$p = Dt/T$ ist die Wahrscheinlichkeit, dass ein Koordinator sich in einem Intervall Dt zurücksetzt, während einer Lebenszeit von T. Die Wahrscheinlichkeit $P[k]$ in der k aus m Koordinatoren im selben Intervall sich zurücksetzen ist dann: $P[k] = \left(\frac{m}{k}\right) p^k (1-p)^{m-k}$

Wenn f Koordinaten sich zurücksetzen, dann wird durch die Minderheit der nicht fehlerhaften Koordinaten, die Korrektheit des Voting-Mechanismus verletzt. Dies tritt ein, wenn $m - f \leq N/2$ oder in anderen Worten $f \geq m - N/2$. Die Wahrscheinlichkeit, dass der Mechanismus verletzt wird, ist dann folgende $\sum_{k=m-N/2}^{m} P[k]$.

Wenn Prozesse beim Anfragen einer Ressource abgelehnt werden, wird die Anfrage fallen gelassen und es wird später versucht. Versuchen allerdings viele Prozesse auf dieselbe Ressource zuzugreifen, so sinkt die Nutzung rapide. In diesem Fall kann es dazu führen, dass genug Prozesse um eine Ressource konkurrieren, dass kein Vote mit einer Mehrheit zustande kommt, sodass die Ressource am Ende ungenutzt bleibt. Die Lösung des Problems ist in der Quelle (S.-D., et al., 2004) zu finden. [59]

[59] vgl. aus dem Englischen (Tanenbaum & Steen, 2018)

4 Einordnung der Technologien anhand von definierten Kriterien

In diesem Kapitel sollen die hier zu diskutierenden Technologien anhand der definierten Kriterien eingeordnet werden. Es werden Argumente aufgeführt, die zu einer positiven Bewertung der Kriterien beitragen soll. Es beginnt mit der PPB, darauf folgt das zentrale Datenbanksystem, das verteilte Datenbanksystem und zuletzt das Cloud-Datenbanksystem.

4.1 Einordnung der PPB anhand definierten Kriterien

In diesem Abschnitt werden Argumente für die Technologie hinsichtlich der Skalierbarkeit, Netzwerksicherheit, Informationssicherheit und der Dezentralität gesammelt und vorgestellt. Hierbei werden die Argumente wertfrei dargestellt, um im späteren Kapitel interpretiert und bewertet zu werden.

4.1.1 Skalierbarkeit

Größen-Skalierbarkeit einer PPB

Um die Größen-Skalierbarkeit in einem Blockchain-Netzwerk zu verstehen, muss die Infrastruktur hinter einer Blockchain vorgestellt werden.

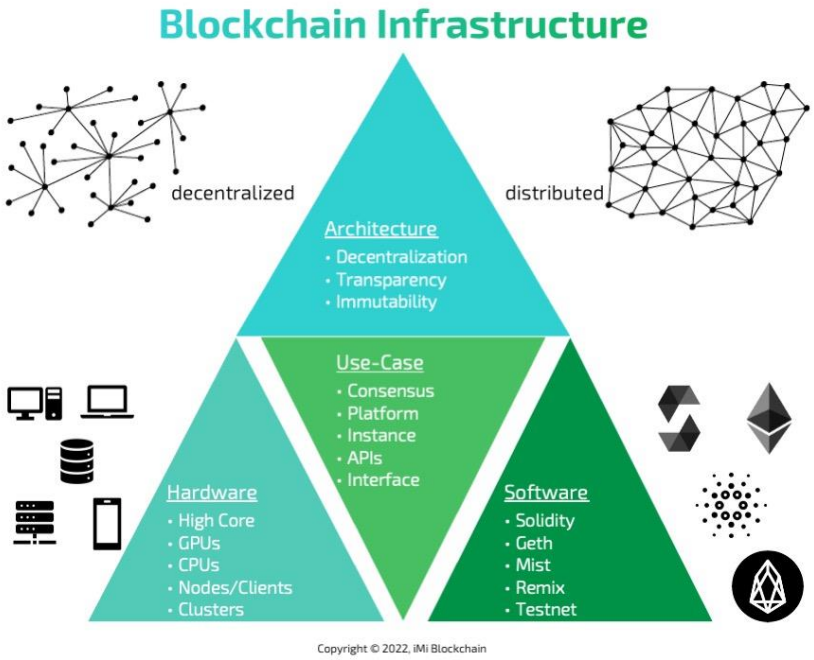

Abbildung 2.5 - Blockchain Infrastruktur[60]

Entscheidet ein Unternehmen sich dafür, eine Private-Blockchain zu implementieren, müssen zunächst die grundlegenden Pfeiler eines jeden Informatiksystems bedacht werden. Es bedarf, wie in Abb. 2.5 dargestellt, erst eines entsprechenden Use Cases, in welchem die Blockchain eine Lösung anbieten kann. Auf Basis dessen wird eine Architektur erarbeitet, die auf das Unternehmen zugeschnitten ist. Als Lösung schlägt diese Arbeit die Implementierung der Private Permissioned Blockchain vor. Was das für eine Blockchain bedeutet, wird in Kapitel 5.2 erläutert. Sind der Use Case und die Architektur bekannt, werden für die weitere Umsetzung Hardware und Software benötigt. Die Blockchain-Entwicklung basiert auf Open Source Code. Wird die Blockchain für eine Unternehmenslösung entwickelt und der weiterentwickelte Code soll nicht allen zur Verfügung stehen, spaltet man sich von der Open Source Community ab

[60] (Isler, 2022)

und erzeugt einen Fork[61].[62] Dies ist auf der Ethereum Plattform einige Male passiert. Herbeigeführt unter anderem durch Hackangriffe oder durch Experimente mit Ethereum, die eine Erhöhung der TPS zum Ziel haben.[63]

Bei Plattformen ist darauf zu achten, wie die Blockchain einzusetzen ist. Ethereum hat den Anspruch völlig dezentral zu sein und unabhängig von dritten Zahlungsinstanzen eigene dApps[64] entwickeln zu können.[65]

Hyperledger-Fabric ist eine Plattform, die sich darauf spezialisiert hat, Unternehmen ein Netzwerk zu gewähren, in dem Unternehmen untereinander Informationen übermitteln können. Nicht jedem Teilnehmer im Netzwerk ist ersichtlich, welche Unternehmen untereinander Informationen und Daten austauschen.[66] Für die Implementierung der Private Permissioned Blockchain schlägt diese Arbeit die Hyperledger-Fabric Plattform als Basis vor.

Diese Plattform stellt Tutorials und Dokumentationen zur Verfügung, wie eine Private-Blockchain erstellt werden kann. Aus diesen Dokumentationen werden die System Requirements für eine Private Blockchain entnommen, welche allerdings von unterschiedliche Faktoren abhängen sind. Unter anderem geht es hierbei um die eingesetzte Block-Größe und die Komplexität, der einzusetzenden Protokollen sowie um die folgende Fragestellung: Wie groß soll das Netzwerk auf der Welt sein? Wie viele Transaktionen sollten auf der Plattform verarbeitet werden können? Wurde dies entschieden, empfiehlt die Hyper-Ledger-Plattform dem Rechner (Node) einen Arbeitsspeicher von acht GB für die lokalen virtuellen Server und eine SSD für eine

[61] Ein Fork ist eine Abspaltung, in der zunächst eine Kopie vom Code einer bestimmten Version gemacht wird. Auf dieser Kopie kann der Code für ein anderes Projekt weiterentwickelt werden, als es für das ursprüngliche Mutterprojekt vorgesehen war.

[62] vgl. (Isler, 2022)

[63] Ethereum Fork Guide: All You Need To Know About Ethereum Hard Fork (bitdegree.org)

[64] dezentrale Applikationen

[65] vgl. (Buterin, 2020)

[66] vgl. (Hyperledger Fabric, 2017)

höhere Performance. Sind die Nodes nur lesend im Netzwerk und beteiligen sich nicht aktiv am Konsensmechanismus, können die Anforderungen angepasst werden. [67]

Um die Infrastruktur abzuschließen, kann zum Entwickeln der Blockchain Tessera verwendet werden, welches auf Java basiert.[68] Die Datenbank, die zunächst nur die Transaktionshistorie verwaltet, ist das Distributed Ledger.[69]

Geographische Skalierbarkeit der PPB

Um eine Geographische Skalierbarkeit zu erreichen, sind Problemen, wie einer verzögerten Transaktion, entgegenzuwirken. Dort gelten die Argumente aus dem Blockchain Trilemma im Abschnitt 3.2. Die Informationen in den Blöcken selbst sind nur wenige Megabytes groß. Der Bottleneck ist nicht die Blockgröße, sondern das Validieren der Blöcke. Bei dem Bitcoin wird alle Zehn Minuten eine Transaktion verifiziert. Die Verzögerung einer Transaktion ist der erhöhten Kryptografie des Bitcoins geschuldet.

Andere Plattformen wie Ethereum oder Hyperledger-Fabric bieten die Möglichkeit Daten und Informationen zu separieren. Dies findet unter dem Begriff Off-Chain statt. In einem Diskussionspapier der TU Berlin (Eberhardt & Tai, 2017) sind Fünf Off-Chain Patterns beschrieben. Eine davon wird als Argumente für die Geographische Skalierbarkeit der Blockchain herangezogen.

Off-Chain Signature Pattern

Es wird davon ausgegangen, dass zwei Netzwerkteilnehmer im Voraus wissen, dass eine Transaktion durchgeführt wird. Sie wollen die Kosten der Transaktion reduzieren oder von anderen Netzwerkteilnehmern verstecken. Dafür ergibt sich folgende Lösung: Die beiden Netzwerkteilnehmer einigen sich auf einen Smart-Contract welcher die Codefunktion besitzt, einen externen Zustand, als Argumente auf den Vertragszustand anzuwenden. Der Smart-Contract überprüft die Signatur der beiden Teilnehmer und stellt sicher, dass beide der

[67] vgl. (Hyperledger Besu, 2022)

[68] vgl. (Quorum, 2022)

[69] vgl. (Hyperledger Fabric, 2017)

Zustandsänderung zugestimmt haben. Nur, wenn beide Teilnehmer zugestimmt haben, dass ein neuer Zustand gewünscht ist, kommt dieser Vertrag zustande und die Transaktion wird auf der Blockchain abgelegt. Daraufhin wird eine Off-Chain-Verbindung zwischen den Peers aufgebaut. Ein Teilnehmer verpackt seine Daten, signiert, dass sich der Zustand der Daten ändert und schickt sie dem anderen Peer zu. Dieser überprüft die Signatur, bestätigt diese und signiert die Zustandsänderung der Daten. Ist die Transaktion von beiden Parteien unterschrieben, kann sie später dem Smart-Contract angefügt werden.[70]

Ein weiteres Beispiel für eine Off-Chain kann dem Konferenzpapier von (Xiao, et al., 2018) entnommen werden. Darin werden die medizinischen Daten von öffentlich-rechtlichen Organisationen verwaltet, auf der Blockchain (On-Chain) wird ein Hash-Wert des Datensatzes abgelegt für die eine bessere Nachvollziehbarkeit. Anhand einer Benutzeroberfläche können die Ärzte, nachdem die sich registriert haben, einen Antrag auf Daten stellen.[71] Danach ähnelt der Ablauf wie schon im oberen Abschnitt des Off-Chain signature Pattern.

Administrative Skalierbarkeit einer PPB

Für die Private Permissioned Blockchain ist die Administrative-Skalierbarkeit ein wichtiger Aspekt. Die Technologie ermöglicht es, Daten und Informationen unternehmensintern sicher zu transferieren sowie diese mit anderen Unternehmen auszutauschen.[72]

In diesem Fall sind nicht nur die hauseigenen Administratoren für neue Richtlinien zuständig, sondern auch die Unternehmen, die an dem Blockchain-Netzwerk teilnehmen. Für eine effizientere Entscheidung für Richtlinien ist z.B. die Hyperledger Fabric Plattform in der Lage Smart-Contracts zu nutzen. Wie schon in Kapitel 2.1.5 beschrieben, sind Smart-Contracts dafür geeignet, automatisch, selbstständig und deterministisch zu entscheiden.

Für eine entsprechende Administrierung der Blockchain werden verschiedene Listen geführt. Es wird eine Administrator-Liste definiert, darin wird festgehalten welche Admins neue Nodes der Blockchain hinzufügen dürfen. Außerdem eine Liste für die Nodes, die Transaktionen

[70] vgl. (Eberhardt & Tai, 2017)

[71] vgl. (Xiao, et al., 2018)

[72] vgl. (Hyperledger Fabric, 2017)

tätigen und verifizieren dürfen sowie eine Liste mit allen Nodes, die am Netzwerk passiv teilnehmen.[73]

4.1.2 Netzwerksicherheit und Angriffe auf eine PPB

Die gängigsten Angriffe auf eine Blockchain

Sicherheitskriterien für eine Public-Blockchain unterscheiden sich von jenen, die für eine Private-Blockchain relevant sind. Der Vorteil der Public-Blockchain ist die komplette Transparenz der Blockchain durch den öffentlichen Zugang. Durch das Einsetzen einer Währung oder das Belohnen sollen die Angriffe gegen das Netzwerk minimiert werden.[74]

Eine gefährliche Angriffsmöglichkeit in einer Public-Blockchain ist die „51%-Attacke". Diese sorgt dafür, dass eine Gruppe von Minern[75] 51% der Netzwerkressourcen besitzen, um die nächsten Blöcke validieren zu können. Sind sie im Besitz der Ressourcen, könnten schon genutzte Coins doppelt ausgegeben oder die Blockchain könnte vom Rest des Netzwerks isoliert werden. Im Umkehrschluss hat dieser Angriff keine große Bedeutung für eine Private Permissioned Blockchain, da dort nicht jedem Zugang zur Blockchain gewährt wird.[76]

Wie dem Argument aus der Administrativen Skalierung zu entnehmen ist, sind dem Netzwerk alle Teilnehmer über eine Liste bekannt. Sollte ein bekannter Node gegen den Konsensmechanismus agieren, kann dieser gezielt bestraft oder ausgeschaltet werden.

Für den nächsten Abschnitt werden die Zugangsbeschränkungen einzeln betrachtet, aufgezeigt, welche Angriffe gegen eine Permissioned Blockchain existieren und im Anschluss erläutert, wie Angriffe auf eine Private-Blockchain funktionieren.

---

[73] vgl. (Hyperledger Besu, 2022)

[74] vgl. (Buterin, 2020)

[75] Nodes einer Blockchain Plattform, die ihre Rechenkraft zur Verfügung stellen, um die Verschlüsselung der Blöcke zu lösen. Bei erfolgreicher Lösung werden die Miner anteilig belohnt.

[76] vgl. (Ye, et al., 2018)

Zugang zu einer Public Permissioned Blockchain hat jeder, die Validation der Blöcke ist beschränkt. Um zu klären, wie das Vertrauen derjenigen, die als Validierer der Blöcke verifiziert werden, zu gewinnen ist, wird im Folgenden ein Konferenzpapier von A. Davenport et.al. betrachtet. In diesem sind die Angriffe für Public Permissioned Blockchain analysiert worden.

Public Permissioned Blockchain Angriffe

Zu Beginn einer Permissioned Lösung wird ein zentraler Dienst gebraucht, der administriert werden kann und die Nodes für das Netzwerk akzeptiert. Dafür wäre zum Beispiel ein „Membership Service Provider (MSP)" sinnvoll. Diese Dienststelle wird zusammen mit einer Zertifizierungsstelle (CA)[77] geführt. Es gibt für eine MSP Umsetzung viele verschiedene Implementierungen. In diesem Beispiel gibt es eine zentrale MSP pro Blockchain, die administriert werden muss. Um den Zugang in eine Blockchain zu sichern, können Private/Public Keys, Signaturen oder digitale Zertifikate genutzt werden.[78]

Insider Threat

Einen kritischen Angriff aus dem Inneren eines Netzwerkes und/oder einer Organisation, der schwer abzuwehren ist, wird „Insider Threat" bezeichnet. In diesem Szenario wird davon ausgegangen, dass die CA nicht vertrauenswürdig und böswillig dem Netzwerk gegenüber ist. Durch das Infiltrieren der CA geht jegliche Macht über das Kontrollieren der Zugriffe für das Netzwerk, das Hinzufügen von neuen Nodes und Identitäten oder das Akzeptieren von neuen Zertifikaten verloren. Das ist dann möglich, wenn es nur eine zentrale Stelle gibt, die das Netzwerk administriert. Wenn das Permissioned Netzwerk nicht groß genug ist, dann existiert nicht genug Rechenpower, um die Crypto Mechanismen der Blockchain aufrecht zu erhalten. Dieser Mechanismus kann leicht von anderen umgangen werden. Es ist schwierig solch einen Angriff wahrzunehmen, bevor es zu spät ist. Daher sollte diese Möglichkeit niemals vernachlässigt werden. Darüber hinaus ist so ein Eindringen auf verschiedenen Ebenen der

[77] Certificate Authoritys (CA) ist eine Organisation, die digitale Zertifikate herausgibt, um vertrauliche Daten zu verschlüsseln oder sie zu entschlüsseln. Das macht die Organisation, indem sie die Public Keys überprüft und die daraus entstehende Verantwortung für die Zertifikate übernimmt. (Anon., 2022).

[78] vgl. (Hyperledger Fabric, 2017)

Netzwerkarchitektur[79] möglich. Zu nennen wäre hierbei die Applikationsebene, die Netzwerkebene oder die Protokollebene.[80]

Nach der Quelle (A. Davenport, 2018) ist der Angriff sehr kritisch, wenn es nur eine zentrale vertrauenswürdige Stelle gibt. Wird diese Aufgabe auf verschiedene Stellen verteilt, so kann zumindest eine andere CA dafür sorgen, dass die Zertifikate vom kompromittierten System als nicht vertrauenswürdig eingestuft werden.

DNS Attack

Mit jedem neuen Node, welches durch das MSP hinzugefügt wird, gibt es eine Reihe von Angriffspunkten, die mit einem DNS-Angriff möglich sind.[81] Die Nutzung von Services in der Blockchain, wie das Herausgeben von Zertifikaten für neue Nodes, wäre eine DDoS[82] Variante. Dieser Angriff hat meist das Ziel Andere Teilnehmer im Netzwerk mit einer abweichenden IP als DNS zu füttern, um andere Netzwerkservices per DDoS auszuschalten, um so den Ursprung des eigentlichen Angriffs zu verschleiern. [83]

Die erste Lösung wäre, die entsprechenden Ressourcen für den Server durch mehr CPU oder RAM aufzustocken, um den Service stabil zu halten. Durch einen gezielten Angriff auf ein System übersteigt dies jedoch die Kapazitäten jeder Ressource leicht. Demnach ist das Einsetzen einer Anti-DDoS-Appliance eine geeignete Lösung. Mit diesem installierten Schutz wird der eingehende Datenverkehr gefiltert und kann durch altbekannte DoS-Muster neue Filterungen selbst erlernen. Sie ermöglicht ein einfaches Integrieren in die Infrastruktur und hilft dabei, DDoS-Angriffe frühzeitig zu erkennen. Ist das Angriffs-Volumen größer als die verfügbare

---

[79] Referenziert auf das OSI-Schichtenmodell für herstellerunabhängige Kommunikationssysteme (Referenzeinfügen)

[80] vgl. (A. Davenport, 2018)

[81] vgl. M. Saad, J. Spaulding, A. Mohaisen, C. Kamhoua, L. Nijila, and D. H.

Nyang, Exploring the Attack Surface of Blockchain, rep, Definition für DNS-Aplification Attack

[82] Distributed Denial of Service: Ein Angriff, in dem meist ein Bot Netzwerk eine Vielzahl von Anfragen an ein gezieltes Ziel schickt, sodass diese Schnittstelle überlastet und den Service lahmlegt. (Qiao Yan, 2016)

[83] vgl. (A. Davenport, 2018)

Bandbreite, kann der Anti-DDoS-Appliance kaum helfen. Da der Service der Blockchain in einem privaten eingeschränkten Umfeld stattfindet, ist im Best-Case ein Whitelisting der berechtigten Kommunikationspartner die erfolgversprechendste Strategie. [84]

Angriffe auf Private Blockchain

Die größte Gefahr einer Private-Blockchain, egal ob Permissioned oder Permissionless ist der nicht registrierte Zugang in das Netzwerk.[85] Grundlegend sollte erwähnt werden, dass eine Private-Blockchain einem privaten Netzwerk oder dem nicht öffentlichen Netzwerk eines Unternehmens gleicht und damit ähnliche oder gleiche Sicherheitsrisiken entstehen. Da dieses Thema sehr weitreichend ist, werden im Folgenden verschiedene Angriffsbeispiele aus dem Buch von N. Pohlmann: „Cyber-Sicherheit" aus 2019 entnommen und vorgestellt.

Sicherheitsrisiko: Software

In einem privaten Netzwerk gibt es viele Schwachstellen, die meistens durch die Nutzer selbst entstehen. Weiterhin gibt es auch Sicherheitsmangel in der genutzten und entwickelten Software. Durch mangelnden Respekt vor Angriffen oder Zeitdruck entstehen Fehler im Code oder werden erst gar nicht getestet. Solche „Bugs" können erfolgreich für Remote-Angriffe auf das IT-System ausgenutzt werden. Die Ziele der Angriffe sind meistens Sicherheitslücken ausnutzen, um über einen separaten Bot Kanal Informationen über das Netzwerk preiszugeben oder das Netzwerkwerk für weitere Angriffe zu verwenden.

Die Lösung für dieses Problem ist das ausgiebige Testen und Prüfen von Code, unabhängig ob dieser Open-Source ist oder selbst entwickelt wurde. Dies kann und sollte von verschiedenen Teams durchgeführt werden.[86] Die Qualität des Codes in einer privaten Blockchain ist maßgebend und sollte nicht unterschätzt werden. Wie schon im Abschnitt 2.1.5 „Smart-Contracts"

[84] vgl. (Pohlmann, 2019)

[85] Für weitere Details bezüglich Permissioned oder Permissionless siehe Kapitel 2.1.4

[86] vgl. (Daniela Soares Cruzes, 2017)

erwähnt, ist im Code von derzeit eingesetzten Smart-Contracts ein Zugang mit einer Wahrscheinlichkeit von 40 % durch Sicherheitslücken möglich.

Advanced Persistent Threat (APT)

Bei dieser Form des Angriffs wird gezielt gegen ein Opfer-IT-System mit komplexen Angriffstechnologien- und Taktiken vorgegangen. Das Ziel hierbei ist es, Teil des Systems zu werden und solange wie möglich unentdeckt zu bleiben, um möglichst viele Informationen zu sammeln. Sind genügend Informationen gesammelt, werden Unternehmensdaten verschlüsselt und erst gegen Lösegeld wieder freigegeben. Die Mittel dafür sind meist Malware, die sich entweder schon auf dem Computer befunden haben, bevor es dem Blockchain-Netzwerk hinzugefügt wurde oder nachträglich durch Lücken eingeschleust wurden.[87]

Eine gängige Art sich gegen Malware zu schützen ist der sogenannte „Malware Detection as a Service (MDaaS)“. Hierbei wird das Netzwerk auf das Verhalten einer Malware, Malware-Code oder Malware-Berichten analysiert. Dadurch kann Malware leicht erkannt und entschärft werden.[88]

Access Management

Der zuletzt betrachtete Angriff ist der eigene Zugang zum privaten Netzwerk. Es ist eine Registrierung im Netzwerk notwendig. Um Zugang zu erlangen, muss dieser Zugang immer wieder authentifiziert werden. Je nachdem, wie eine Registrierung abgeschlossen wird, erhält der Node einen Key oder besitzt ein Passwort, um sich zu authentifizieren. Es existiert Software, wie beispielsweise ein Keylogger, die die Eingaben der Tastatur abgreifen oder gleich eine Liste mit allen Eingaben erstellen.[89]

---

[87] vgl. (Pohlmann, 2019)

[88] vgl. (Anita N, 2019)

[89] vgl. (Pohlmann, 2019)

Eine Variante, um unautorisierte Zugriffe zu mindern, wäre zum Beispiel eine Zwei-Faktor-Autorisierung. Dabei muss neben der Passworteingabe auch ein Code eingetippt werden, der per Mail oder als SMS zugeschickt wird. [90]

Die genannten Angriffe decken nur einen geringen Anteil des Angriffsspektrums auf ein privates Netzwerk ab. Für weitere Informationen und Varianten wird auf die Quelle von (Pohlmann, 2019) verwiesen.

4.1.3 **Informationssicherheit einer PPB**

Wie im Abschnitt 3.3 „Informationssicherheit" schon erwähnt, gilt es für die Blockchain zu zeigen, wie eine Blockchain sicherstellt, dass die Informationen vertrauenswürdig, verfügbar und deren Integrität gesichert wird.

Zunächst ist für eine Blockchain zu erwähnen, dass die Informationen über Transaktionen aufgrund des Distributed-Ledger gesichert werden, dies wird in Kapitel 2.1 „Herkunft" zuerst erwähnt. Das Distributed-Ledger bringt die Eigenschaft der Unveränderlichkeit mit sich, da das sogenannte Kontobuch die Logs über die Transaktionen nicht löschen soll, sondern nur neue Informationen anhängt. Zusätzlich ist das Distributed-Ledger nicht zentral angelegt, sondern redundant auf jedem Node aufzufinden. Dies funktioniert wie folgt: wird ein Status einer Information verändert und diese Veränderung ist durch den Konsens verifiziert worden, werden alle anderen Nodes im Netzwerk über die Zustandsänderung informiert. Diese Information kann mittels Gossip[91] an die weiteren Nodes verteilt werden.

Im Whitepaper über Bitcoin stellt der Autor (Nakamoto, 2008) dar, dass Transaktionen vom Versender signiert werden. Erhält der nächste Node die signierte Transaktion, kann er über den Public Key die Echtheit der Signatur feststellen. Beim Bitcoin wird die Identität des Urhebers des Public Keys anonym gehalten.

---

[90] vgl. (Rügheimer, 2019)

[91] Methode um Informationen in einem Verteilten Netzwerk automatisch koordiniert zu erhalten oder sie zu verbreiten. (Tanenbaum & Steen, 2018)

Das Verfahren einer digitalen Signatur basiert auf der asymmetrischen Kryptografie.[92] Dies ist ein weit verbreiteter Ansatz, um die Authentizität von Nachrichten oder Informationen zu verifizieren. Dafür ist ein Hash-Verfahren notwendig (siehe Kapitel 2.1.3 Hash-Funktionen). Zuerst wird ein Schlüssel-Paar generiert - eines ist public, das andere ist privat. Der dazugehörige private Schlüssel wird sicher physisch auf einer Festplatte oder Smart-Card aufbewahrt. Der Public Key hingegen kann veröffentlicht werden. Somit wird eine Nachricht mit dem Private-Key signiert und der Empfänger kann mit dem Public Key und dem Hash-Verfahren das Hash der Nachricht entschlüsseln. Stimmt der Hash-Wert mit der geschriebenen Nachricht mit entschlüsseltem Wert überein, stammt die Nachricht tatsächlich vom Sender.[93]

Die Integrität der Blöcke wird durch die digitalen Signaturen und den Smart-Contracts gewahrt. Mittels des Hash-Verfahrens und den Hash-Werten aus Kapitel 2.1.2 „Herkunft" können auch die einzelnen Informationen und Nachrichten aus dem Netzwerk eindeutig überprüft werden.

4.1.4 **Dezentralisierung einer PPB**

Zunächst muss in Bezug auf die Dezentralisierung in einem Private-Permissioned-Netzwerk erwähnt werden, dass die Anforderung zunächst widersprüchlich gesehen werden können. Bei Dezentralisierung, die sowohl horizontal als auch vertikal ist,[94] gibt es mehrere Instanzen auf verschiedenen Ebenen, die dieselben Aufgaben tätigen oder Lösungen für Probleme finden. Daher kann die Dezentralität in einem privat-eingeschränkten Netzwerk in Stufen angenähert werden.

Mit der Eigenschaft von redundanten Admin Nodes, die gemeinsam darüber entscheiden, wer am Netzwerk teilnehmen darf, wie in Kapitel 4.1.1 „Administrative Skalierbarkeit" erläutert,

---

[92] vgl. (Beutelspacher A, 2015)

[93] vgl. (Meier & Fill, 2020)

[94] Der bundesstaatliche Aufbau Deutschlands kann als vertikale Dezentralisierung aufgefasst werden [...] Bund, Länder und auch die Kommunen jeweils eigene Verwaltungsträger, die im Normalfall innerhalb der verfassungsrechtlichen Kompetenzordnung fachweisungsfrei agieren (Art. 83 GG). Die horizontale Dezentralisierung. zeigt sich innerhalb einer der föderalen Ebenen, sofern hier etwa Selbstverwaltungseinheiten geschaffen werden Art. 87 Abs. 2 GG – funktionale Selbstverwaltung der Sozialversicherungsträger auf Bundesebene [...]. Dezentralisation – Staatslexikon (staatslexikon-online.de)

erhält die Blockchain dezentrale Werte. Daraus ergeben sich drei Gruppen aus Nodes, die in den Gruppen dieselbe Arbeit leisten: passive Nodes, die nur den Service der Blockchain nutzen, aktive Nodes, die den Konsensmechanismus der Blockchain durchführen und Admin Nodes, die Blockchain verwalten.[95]

Durch das Distributed-Ledger erhält die Blockchain weitere dezentrale Werte, da auf jedem Node dieselben Datenbankinformationen liegen, was auch als Argument für die Informationssicherheit in Kapitel 4.1.3 genutzt wurde.

4.2 Client/Server Architektur (Zentrales Datenbanksystem)

4.2.1 Skalierbarkeit eines zentralen Datenbanksystem

Mit dieser Datenbank-Architektur ist die Infrastruktur eine ähnlich wie die einer Blockchain, es bedarf ein Use Case sowie eine Architektur des Datenbanksystems. Zusätzlich zu der Architektur des gesamten Systems braucht es eine Datenbankarchitektur. Denn anders als in der Blockchain, in der nur Transaktionen abgespeichert werden, sollen hier strukturierte Informationen eine Relation erhalten.

Dem eingesetzten Server obliegt eine große Rechenkraft, um die Anfragen des DBMS verarbeiten und die strukturierten Daten und Information speichern zu können. Viele der Clients oder Applikationen erhalten meist nur durch ein DBMS System Zugriff auf die Datenbank, da Clients selbst kaum Informationen oder Daten abspeichern, weshalb dafür ein gängiger Laptop aus 2021 ausreichen würde.[96] Die Problematik bei einem zentralen Server ist, die hohe Anzahl an Anfragen, die von einem Server verarbeitet werden muss.

Sowohl die physikalischen Ressourcen als auch das Design der Datenbank sind signifikant. Für zentrale Datenbanken gibt es simple, weniger komplexe Ansätze, da Informationen und Daten nicht mehrfach an verschiedenen Orten existieren und somit nicht repliziert sind. Jedoch haben

[95] vgl. (Abadi & Brunnermeier, 2022)

[96] Anhand den Spezifikationen eines Lenovo IdeaPad 3 14. am 05.2022 festgestellt.

Datenbanken das Problem eines Mehrbenutzer-Zugriffs[97]. Dafür gibt es vier verschiedene Phänomene, die alle durch optimistische sowie pessimistische Sperren und Systemprotokolle behoben werden können.[98] Daher reicht die Architektur einer zentralen Datenbank mit einer LAN- und WAN-Reichweite[99] für die Klein- und Großunternehmen aus.

Erlebt ein Unternehmen einen massiven Gewinn an Kunden und Mitarbeitern, die an unterschiedlichen Standorten tätig sind, reicht eine zentrale Datenbank-Architektur meist nicht aus. Nutzer beziehungsweise Applikationen müssten aufgrund von Mängeln der Netzwerkinfrastruktur mit Delays arbeiten. Der Vorteil einer Client/Server-Architektur ist, dass sie mit wenig Aufwand zu einer verteilten Datenbank weiterentwickelt werden kann.

Geographische Skalierbarkeit zentrale Datenbank

Bei einer Serverarchitektur mit einer zentralen Datenbank würde eine geographische Skalierbarkeit nur dann erreicht werden, wenn eine horizontale Skalierung umgesetzt wird. Dafür müssten dem Netzwerk weitere Ressourcen zur Verfügung gestellt werden. Diese Rechner oder Server würden zum Beispiel die Arbeit und Prozesse der zentralen Datenbank abnehmen.[100]

Administrative Skalierbarkeit

Eine Client/Server Architektur mit einer zentralen Datenbank hat meist wenige Administratoren, die Server sind an einem zentralen Ort aufgestellt. Der Verwaltungsaufwand für eine zentrale Datenbank geht mit der Verwaltung der Zugänge für die Datenbank einher. Es müssen neue Benutzer berechtigt werden oder alte deaktiviert werden. Für diesen Vorgang könnte ein Identity Management Abhilfe schaffen, wobei eine zentrale Speicherung aller Benutzerdaten notwendig ist. Damit können die Zugriffsberechtigung aller Mitarbeiter über das Identity-Management-System verwaltet werden.[101]

---

[97] Mehrere Transaktionen greifen parallel auf dieselbe Information zu.

[98] vgl. (Dadam, 2013)

[99] Local Area Network und Wide Area Network

[100] vgl. (Mock, et al., 2014)

[101] vgl. (Merkel, 2009)

4.2.2 Netzwerksicherheit eines zentralen Datenbanksystem

Für eine zentrale Datenbank, die auf einem Server installiert ist, gibt es, wie schon im Abschnitt 4.1.2 beschrieben, ein ähnliches Angriffsspektrum. Jedoch sollen in diesem Abschnitt die Angriffe hervorgehoben werden, die sich speziell auf die Datenbank beziehen und nicht auf das gesamte Netzwerk.

Database Injection Attack

Für diesen Abschnitt werden Ausschnitte über den Angriff und die Verteidigungsmöglichkeiten aus dem Buch „SQL Injection Attacks und Defense“ von (Clarke, 2009) hinzugezogen.

Zunächst wird erklärt, was ein „SQL Injection Attack“ ist. Hierbei handelt sich um einen Angriff, in dem der Angreifer Zugriff auf das Netzwerk erhalten hat und über SQL-Queries versucht, Informationen aus einer Datenbank zu erhalten. Diese Informationen können Benutzernamen, Passwörter, Namen und Adressen oder Telefonnummern sein. Diese Art des Angriffs kommt meistens dann zustande, sobald eine Datenbank mit einer Webseite verknüpft ist. Es gibt einen direkten und einen indirekten Angriff. Beim direkten Angriff wird versucht, durch ein Feld der Webseite, die auf die Eingabe des Users wartet, den SQL-Befehl zu übermitteln. Die Eingabe des Users wird an die Back-End Datenbank geschickt und führt den eingegeben Befehl aus, der dann eine Antwort liefert. Bei einem indirekten Angriff wird der bösartige Code irgendwo mit anderen Informationen platziert und wartet darauf, dass diese Information in der Datenbank in einer Tabelle gespeichert wird, um den Code dann auszuführen. Dabei besteht die Gefahr, dass sofern der Zugriff über diese Datenbank gewährt worden ist, die Angreifer dieselben Rechte wie die Applikation und die Datenbank besitzen. Zudem können diese dann mit anderen Schnittstellen innerhalb des Netzwerks kommunizieren.[102]

Für eine erfolgreiche Verteidigung wird eine Möglichkeit zum Umsetzen vorgestellt, diese wird auf der Code-Ebene durchgeführt.

[102] vgl. (Clarke, 2009)

```
public boolean isValidPassword(String username, String password)
{
   String sql = "SELECT * FROM user WHERE username='" + username + "'
   AND password='" + password + "'";
   Result result = query(sql);
   ...
}
```

Abbildung 2.6 - Code Beispiel für generische Argumente[103]

Auf der Code Ebene gibt es die „Domain Driven Security (DDS)“. Dies ist eine Zustimmung, dass Code so entwickelt wird, dass typische Injection Angriffe verhindert werden können. Die DDS schreibt zum Beispiel vor, dass Argumente im Code nicht generisch sein dürfen. Im Abbild 4.2.2 ist ein Beispiel-Code, der für eine Webseite implementiert wurde. Problematisch an diesem Code ist, dass wenn der User eine Eingabe tätigt und diese bspw. 40 Zeichen lang ist, dies einfach akzeptiert wird. Das kann dazu führen, dass in so einem Eingabefeld ganze SQL-Queries eingefügt werden können. Somit sollten die erlaubten Zeichen eingeschränkt werden, sodass kein gültiger Befehl zustande kommen kann.

[103] (Clarke, 2009)

```
public class Username {
   private static Pattern USERNAME_PATTERN = Pattern.compile("^[a-z]
   {4,20}$");
   private final String username;
   public Username(String username) {
      if (!isValid(username)) {
         throw new IllegalArgumentException("Invalid username:"
         + username);
      }
      this.username = username;
   }
   public static boolean isValid(String username) {
      return USERNAME_PATTERN.matcher(username).matches();
   }
}
```

Abbildung 2.7 – Code, welcher nach dem DDS Prinzip geschrieben wurde (Clarke, 2009)

Abbild 2.7 zeigt einen Code, bei dem ein Injection-Angriff in der direkten Form keine Auswirkung hätte. Das Argument wird nicht direkt in eine SQL-Query geschrieben, sondern zunächst als eine eigene Klasse erzeugt. Hierbei wird nach der Eingabe geprüft, ob das Argument 4-20 Zeichen lang ist und nur Zeichen von A bis Z benutzt. Erst wenn diese Prüfung getätigt wurde, wird die Eingabe als Query in die Datenbank geschickt. [104]

Angriff auf nicht aktuelle Datenbanksysteme

Dieser Angriff zielt auf veraltete Server im Netzwerk, es wurden Lücken im System vom Entwickler entdeckt, die wiederum von Angreifer ausgenutzt werden können. Daher ist die einfache Lösung, seine Systeme stets Up-to-Date zu haben. Um die gefundenen Lücken zu schließen.[105]

[104] vgl. (Clarke, 2009)

[105] vgl. (TRUST 2014, 2014)

4.2.3 **Informationssicherheit eines zentralen Datenbanksystem**

Bei der Entwicklung eines Datenbanksystems wird sich in den meisten Fällen an den fundamentalen Eigenschaften von dem Forscher Edgar Frank Codd orientiert. Dazu werden im Folgenden die neun Codd'schen Regeln vorgestellt, die unter anderem dafür sorgen, dass die Kriterien für die Informationssicherheit erfüllt werden. Diese Regeln gelten für Datenbanken im Allgemeinen.

1. Integration (einheitlich, redundanzfrei), alle Daten werden von einem Datenbank-Managementsystem verwaltet.
2. Operation (Speichern, Ändern, Löschen, Lesen), dies sind Zugriffsoperationen, die auf einem Datenbanksystem möglich sein sollen.
3. Katalog (Zugriff auf Datenbankbeschreibungen im Data Dictionary), eine Übersicht, welche Datenstruktur es gibt und wie sie aufgebaut ist.
4. Benutzersichten (Benutzerspezifische Sicht auf Relationen), im Katalog ist auch gespeichert, wer Zugriff auf welche Daten hat. Es soll möglich sein, Benutzern unterschiedliche Rechte zu vergeben.
5. Integritätssicherung (Korrektheit des Datenbankinhalts), im DBMS können Regeln für Daten aufgestellt werden, sodass keine inkonsistenten oder falsche Informationen gespeichert werden.
6. Datenschutz (Verhinderung von unautorisiertem Zugriff), das DBMS sorgt dafür, dass nur die Benutzer Zugriff auf das System erhalten, die auch dafür berechtigt sind.
7. Transaktion (mehrere DB-Operationen als eine logische Einheit definierbar), hierbei werden unterschiedliche Operationen verknüpft, um keine inkonsistenten Daten zu generieren. Zum Beispiel wird eine Änderung an einem Datum erst dann gespeichert, wenn das DBMS keine Operation mehr auf das Datum feststellt (Alles oder nichts Prinzip).
8. Synchronisation (parallele Transaktionen koordinieren), dabei achtet das DBMS darauf, dass nicht mehrere Benutzer dieselben Daten verändern, sondern mit Sperren arbeitet.

9. Datensicherung (Wiederherstellung von Daten nach Systemfehlern), dies ermöglicht ein Wiederherstellen von Daten nach Hard- und Softwarefehlern, sodass die Datenbank wieder konsistent ist. [106]

Damit ist die Verfügbarkeit und die Integrität der Daten für den Benutzer gewährleistet. Jedoch ist die Verfügbarkeit nur so weit verfügbar, wie der physikalische Server es auch ist. In der Diplomarbeit von Thomas Hertz aus dem Jahr 2003, wird gezeigt, dass performante Mechanismen wie das Verteilen auf verschiedene physikalische Server notwendig sind, um die Skalierbarkeit und die Ausfallsicherheit zu erhöhen. [107]

Für die Vertrauenswürdigkeit kann der Kommunikationskanal mit der Datenbank Ende-zu-Ende verschlüsselt werden. Damit können Dritte, Informationen und Daten nicht mitlesen oder manipulieren.[108]

4.2.4 Dezentralität eines zentralen Datenbanksystem

Für eine zentrale Datenbank ist eine Dezentralität meist nicht gewünscht, wenn jedoch die Argumente aus Kapitel 4.2.1 „Skalierbarkeit“ der DBMS hinzugezogen werden. So hat eine zentrale Datenbank eine geringfügige Dezentralität durch die DBMS-Schnittstelle.

4.3 Verteiltes Datenbanksystem

4.3.1 Skalierbarkeit eines verteilten Datenbanksystem

Größen-Skalierbarkeit

Mit einer Erweiterung der Client/Server-Architektur durch zusätzliche Server wird es zu einem Mulit-Client/Multi-Server System. Damit können in erster Linie einige Probleme einer Zentralisierung reduziert werden. Unternehmen mit einem zentralen Server wären bei Systemausfällen durch Netzwerkstörungen oder Gebäudebrand nicht arbeitsfähig. Durch das redundante

[106] vgl. (Gerken, 2016)

[107] vgl. (Hertz, 2003)

[108] vgl. (Bundesamt für Sicherheit in der Informationstechnik, 2020)

Verteilen der Datenbank auf verschiedene Standorte ist es im Falle solcher Ausnahmesituationen weiterhin möglich, Zugriff auf Informationen zu gewähren. Dies ist aus den Vorteilen der Tabelle im Abschnitt 2.2 zu entnehmen.

Durch die zusätzlichen eingesetzten Ressourcen ist es dem verteilten System möglich, nach dem Scaleup-Prinzip zu agieren. Die Antwortzeit der Systeme würde sich mit zusätzlichen Benutzern und Schnittstellen verschlechtern, allerdings nicht, wenn die Hardware-Ressourcen mitwachsen. Daraus entsteht die folgende Formel:

$scaleup = {volume_{parallel}}/{volume_{original}}$. $volume_{parallel}$ entspricht der Transaktion in einer bestimmten Zeit (TPS). Um die TPS zu erreichen, unterstützen dabei die DBMS-Systeme. Die sind so angepasst, dass es zunächst einen Home-Server gibt, wo die notwendigen Informationen abgefragt werden. Sollten die Informationen nicht auf dem Server enthalten sein, wird die Anfrage auf eine der anderen Datenbanken weitergeleitet.[109]

Geographische Skalierbarkeit

Durch den Einsatz verschiedener Serverstandorte sind verteilte Datenbanken kompatibel, mit einer größeren Geographischen Skalierbarkeit.[110]

Administrative Skalierbarkeit

Wenn zusätzliche Server an unterschiedlichen Standorten eingesetzt werden, steigt der Verwaltungsaufwand für die Administratoren. Bei einem verteilten Datenbanksystem möchte man eine lokale Unabhängigkeit erreichen. Jedes der Serverstandorte soll für sich selbst autonom funktionieren, sodass keine Abhängigkeiten zu anderen Servern oder Seiten bestehen, die nicht selbst verwaltet werden.

Wenn ein verteiltes System eingesetzt wird, ergeben sich Möglichkeiten, die Daten aufzuteilen, durch sogenanntes Fragmentieren. Diese Fragmente können weltweit unter den Datenbanken

[109] vgl. (Ray, 2009)

[110] vgl. (Tanenbaum & Steen, 2018)

aufgeteilt werden. Damit diese Fragmente wieder zusammengeführt werden können, braucht es ein optimal verteiltes System - eine Allokation[111]. Um die Verfügbarkeit der Fragmente zu gewährleisten, bedarf es Replikationen.[112]

4.3.2 Netzwerksicherheit eines verteilten Datenbanksystem

Die schon vorgestellten Methoden für zentrale Systeme und der Blockchain Technologie können auch für das verteilte System angewendet werden. Mit einer verteilten Datenbank und ein daraus resultierendes verteiltem System entstehen neue Probleme für die Netzwerksicherheit. Diese Arbeit wird lediglich ein paar Argumente über dieses sehr weitreichende Thema liefern, diese stammen aus dem Buch von „Distributed Systems" (Tanenbaum & Steen, 2018).

Authentifizierung ist die wohl effektivste Art, ein verteiltes Netzwerk zu schützen. Ohne Sicherstellung der Integrität ist die Authentifizierung hinfällig. Ein Anwendungsbeispiel wäre: Bob kann durch den Usernamen nachvollziehen, dass er mit Alice kommuniziert. Dennoch ist es fragwürdig, ob die Nachricht tatsächlich von Alice stammt. Für diese Art des Problems könnte ein Session-Key eingesetzt werden. Dies ist ein gemeinsam genutzter (geheimer) Schlüssel, mit dem es möglich ist, die verschlüsselten Nachrichten zu entschlüsseln und somit die Integrität und Vertraulichkeit zu wahren. Nachdem der Kommunikationskanal nicht mehr genutzt wird, kann der Schlüssel weggeworfen werden.

Problem an dieser Methode ist die Skalierbarkeit. Ist eine hohe Anzahl an Nutzer N vorhanden, so muss das System $N(N\text{-}1)/2$ Schlüssel verwalten und jeder Host $N - 1$ Schlüssel.[113]

4.3.3 Informationssicherheit eines verteilten Datenbanksystem

Wie im Abschnitt 4.3.2 „Netzwerksicherheit verteiltes Datenbanksystem" bereits der Session-Key vorgestellt wurde, so erfüllt diese Methode auch die Kriterien für eine erfolgreiche Informationssicherheit bezüglich der Integrität und der Vertraulichkeit der Daten.

---

[111] ermöglicht das Zuweisen von Fragmenten

[112] vgl. (Ray, 2009)

[113] vgl. (Tanenbaum & Steen, 2018)

Wie dem Abschnitt 4.3.1 „Skalierbarkeit verteiltes Datenbanksystem“ zu entnehmen ist, ermöglichen DDBMS[114] eine entsprechende Verfügbarkeit durch die Multi-Client/Multi-Server Architektur.

4.3.4 Dezentralität eines verteilten Datenbanksystem

Ein verteiltes Datenbanksystem übernimmt die erreichte Dezentralität eines zentralen Datenbanksystems durch die einfache Erweiterung der Client/Server-Architektur. Zusätzlich gewinnt ein verteiltes Datenbanksystem durch die lokale Unabhängigkeit der Serverstandorte an Dezentralität. Durch die vorhandenen Replikationen der Fragmente und Datenbanken ist es möglich, unabhängiger vom Gesamtsystem zu funktionieren (Zuerst erwähnt im Abs. 4.3.1 „Skalierbarkeit verteiltes Datenbanksystem“). Dies reicht allerdings nicht, um vollständig dezentral zu sein. Auf den einzelnen Endgeräten wirkt die Software so als ob sie eigenständig den Service abwickelt, jedoch werden die Serviceanfragen an eine Middleware weitergeleitet und von einem Server bearbeitet. [115]

4.4 Cloud-Datenbanksystem

4.4.1 Skalierbarkeit eines Cloud-Datenbanksystem

Größen-Skalierbar

Mit diesem Datenbankmodell können Nutzer Ressourcen flexibel nutzen. Eine Cloud-Datenbank kombiniert eine zentrale Serverarchitektur mit verteilten Elementen. Die Daten sind vollständig an einem Ort und müssen nicht auf andere Datencenter oder Standorte verteilt oder fragmentiert werden. Die Kommunikation zur Datenbank ist dann gewährt, sobald man Zugriff auf das Internet hat. Sollte die Datenbank für zukünftige Erweiterungen größer werden, ist es

[114] Distributed Database Management System

[115] (Tanenbaum & Steen, 2018)

möglich, zusätzlichen Speicher zu nutzen. Werden die Ressourcen nicht mehr benötigt, so können diese wieder abbestellt werden. [116]

Für die unterschiedlichsten Anforderungen wird ein Service angeboten, sodass der Nutzen und die Skalierbarkeit von finanziellen Ressourcen abhängig ist. Bei ca. 4,9 Milliarden Menschen, die Zugang zum Internet haben,[117] kann auch die Cloud-Architektur nicht allen gerecht werden, jedoch besitzt die Architektur die richtige Infrastruktur, um über 9 Millionen Webseiten zu hosten.[118]

Geographische Skalierbarkeit

In Cloud-Datenbanken können jegliche Arten von Verwaltung als Service gebucht werden. Je nach Modell kann eine Applikation erworben werden, die bereits eine fertige Datenbank nutzt und nur ein Frontend benötigt wird. Diese Datenbankarchitektur wird als „Everything-as-a-Service" vermarktet. Bei einer öffentlichen Cloud-Datenbank ist der weltweite Zugang möglich und eine Cloud-Lösung bekommt GAN[119] Reichweite. Dabei muss der Betreiber des Cloudservice lediglich dafür sorgen, dass die eigenen Kommunikationsschnittstellen verfügbar sind und bleiben, um den vollen Service zu gewährleisten. [120]

Administrative Skalierbarkeit

Die Administration der zentralen Serverarchitektur und der Cloud-Architektur ähneln sich. Der administrative Aufwand der Cloud-Architektur ist jedoch ein umfangreicher, da die Administratoren eine ganze Serverfarm verwalten. Je nachdem, wie viele Server für Kunden verwaltet werden müssen, werden auch entsprechende Administratoren für die Services eingesetzt.

---

[116] vgl. (Shehri, 2013)

[117] vgl. (L. Rabe, 2021)

[118] (kinsta, 2022)

[119] Global Area Network

[120] vgl. (Pizette & Cabot, 2012)

Anders als bei den üblichen Datenzentren, die ein Unternehmen im Hause hat, wird jeder Service bezahlt. [121]

Ein regelmäßiger Verwaltungsaufwand für Cloud-Servicebetreiber ist der Austausch von Festplatten. Es wird mit einer jährlichen „Hard Drive Failure-Rate“ von ca. 1 % gerechnet. Bei Backblaze[122], die im ersten Quartal 2022 207.478 Festplatten eingesetzt haben, mussten 619 fehlerhafte Festplatten ausgetauscht werden.[123]

4.4.2 Netzwerksicherheit eines Cloud-Datenbanksystem

Durch den Zugang über Internetschnittstellen und den erhöhten Nutzen von Großunternehmen, ist die Clouddatenbank besonders für DDoS-Angriffe anfällig. (Im Abschnitt 4.1.2 „Netzwerksicherheit der Blockchain“, wird der Angriff näher erläutert.) Weitere Verteidigungsmöglichkeiten für DDoS-Angriffe in einer Cloudumgebung sind dem Papier von (Qiao Yan, 2016) zitiert nach (Zargar, et al., 2013) zu entnehmen.
Diese werden in 4 Kategorien eingeteilt:

- Quellenbasierter Mechanismus: Hierbei wird an der Quelle der Angriffsmöglichkeit direkt angesetzt und am Router schon gefiltert.
- Netzwerkbasierter Mechanismus: Dafür werden die Mechanismen auf dem Router oder im Netzwerk platziert und sollen helfen, DDoS-Angriffe frühzeitig zu entdecken.
- Zielbasierter Mechanismus: Dafür setzt der Mechanismus beim Empfänger des Angriffs an, mittels debuggen oder hash-basierter IP-Rückverfolgung.
- Hybridmechanismus: Zu diesem Zweck wird der Mechanismus an verschiedenen Orten eingesetzt, wie zum Beispiel, an der Quelle, im Netzwerk oder beim Empfänger. [124]

---

[121] vgl. (Pizette & Cabot, 2012)

[122] Ein US-Amerikanisches Unternehmen welches 4 Datencenter verwaltet, 3 in den USA und 1 in Europa, sie bieten Cloud Speicher und Datenbackups an. (Sanders, 2019)

[123] (Blackblaze, 2022)

[124] vgl. (Qiao Yan, 2016)

Laut eines Unternehmensberichts von bitkom, indem verschiedene Unternehmen zu Cloud-Lösungen befragt worden sind, zeigte sich, dass 2021 im Vergleich zum Vorjahr mehr Unternehmen eine Cloud-Lösung nutzen. Dieser Umfrage nach befürchten die meisten Firmen einen unberechtigten Zugang auf sensible Daten. Es würden über 50 % unautorisierte Zugriffe auf den Public-Cloud-Dienst registriert werden. Zudem seien knapp 20 % der Meinung, es fehle die Möglichkeit für eigene Audits.[125]

4.4.3 Informationssicherheit eines Cloud-Datenbanksystem

Die Informationssicherheit für eine Cloud-basierte Lösung stellt eine andere Problematik als bei den bisherigen Technologien dar. Wenn eine Datenbank als Service gebucht wird, so ist man nie selbst im Besitz der Daten. Dennoch können die Kriterien der Informationssicherheit erfüllt werden, dafür werden die Argumente aus dem Papier von (Hansen, 2012) herangezogen.

Für eine sichere Datenübertragung zwischen Anwender und Cloudanbieter kann die Vertraulichkeit mittels SSL erfüllt werden. Ein weiteres Hilfsmittel wäre das Fragmentieren von Daten, damit könnten Teile der Daten in einer privaten Cloud abgelegt werden und andere Teile in der Public Cloud.

Um die Integrität der Daten zu gewährleisten, wie schon in Abschnitt 4.1.3 erwähnt, können ebenfalls Prüfsummen oder Signaturen verwendet werden.[126]

4.4.4 Dezentralität eines Cloud-Datenbanksystem

Die Clouddatenbank ist nicht in der Lage eine Dezentralität zu leisten, da der Client nicht die Rolle des Servers einnehmen kann. Durch ein DBMS können Teile der Arbeit übernommen werden, allerdings sind dies Eigenschaften eines verteilten Systems, wie sie dem Abs. 4.2.4 und Abs. 3.6 zu entnehmen sind. Die Clouddatenbank ist durch Fragmentierung in der Lage, die Verantwortung der Public Clouddatenbank auf eine private Clouddatenbank zu verteilen (siehe Abs. 4.4.3 „Informationssicherheit Cloud-Datenbanksystem“).

---

[125] vgl. (Gentemann & Heidkamp, 2021)

[126] vgl. (Hansen, 2012)

5 Erkenntnisse und Diskussion des Vergleichs der diskutierten Technologien

In diesem Kapitel werden die Argumente aus Kapitel 4 zusammengefasst und mit der Bewertungstabelle aus Kapitel 3.1 bewertet, um diese anhand von Tabellen optisch zu vergleichen. Im Anschluss werden die Ergebnisse interpretiert. Daraufhin soll die genutzte Literatur kritisch hinterfragt werden und zuletzt erfolgt aus den gewonnenen Erfahrungen eine praktische Implikation.

5.1 Erkenntnisse aus der Einordnung der Technologien

Anhand der Bewertungstabelle (vgl. Tabelle 2.5.) sind die Kriterien in Stufen erfüllbar. Ist eine Technologie zum Beispiel nicht in der Lage die Dezentralität umzusetzen so wird dies mit nicht vorhanden bewertet, können nur ein Teil der vorgestellten Angriffe verteidigt werden so wird die Netzwerksicherheit nicht vollständig bewertet.

Stufe	0	1	2	3	4
Wertung	nicht vorhanden	gering	mittel	gut	vollständig
Stufe in %	0	25	50	75	100

Tabelle 2.5 - Bewertungstabelle aus Kapitel 3.1

Private Permission Blockchain

Zusammenfassend lässt sich über die PPB sagen, dass die Technologie in ihrer Ausführung sehr komplex, aber flexibel ist. Die Skalierbarkeit wird durch neue Nodes, die zusätzlich neue Ressourcen zur Verfügung stellen, positiv beeinflusst. Zusätzlich trägt eine Implementierung einer Off-Chain dazu bei, dass größere Datenmengen über eine P2P-Verbindung übertragen werden können. Wohingegen der große Verwaltungsaufwand mit wachsenden Nodes und

Unternehmenspartnern durch Smart Contracts einen negativen Einfluss auf die Skalierung hat. Aus den genannten Gründen ist die Skalierbarkeit als „gut“ zu bewerten (vgl. Tabelle 2.5.).

Für die Netzwerksicherheit weist die Blockchain zunächst neue Angriffsmöglichkeiten auf. Diese können durch das Privatisieren und durch das Beschränken der Validierung so weit dezimiert werden, dass lediglich die gängigsten Angriffe auf ein privates Netzwerk bestehen bleiben. In diesem Netzwerk sind die Verteidigungsmaßnahmen implementierbar, wie es für nicht öffentliche Unternehmensnetzwerke ebenfalls möglich wäre. Wenn also die Sicherheitsrisiken, ungetesteter Code und unautorisierte Zugänge, durch ausreichend Tests und einem Access Management beseitigt werden, kann mit einem Konsensmechanismus eine Transaktion zusätzlich geschützt werden. Durch das Einsetzen einer No-SQL Datenbank wird außerdem der SQL-Injection-Angriff verhindert. Der Fokus dieser Blockchain ist nicht, vollkommene Sicherheit zu schaffen. Viel wichtiger ist die Transparenz, einen geregelten und vertrauenswürdigen Austausch von Informationen, weshalb die Bewertung in der Kategorie der Netzwerksicherheit als „gut“ zu bewerten ist (vgl. Tabelle 2.5).

Wie bereits erwähnt, soll eine PPB dazu dienen, nicht vertrauenswürdige Kommunikationspartner Informationen zu liefern, die integer und vertrauenswürdig sind. Dafür sind Mechanismen, wie Smart-Contracts oder dem Distributed Ledger gut geeignet und fester Bestandteil der Blockchain. Dafür, dass jede Transaktion nachvollziehbar bleibt und Regularien automatisch eingehalten werden, sorgen die oben genannten Smart-Contracts und die Distributed Ledger. Informationen, die auf dem Distributed Ledger und in einem Netzwerk mit redundanten Nodes abgelegt sind, sind immer verfügbar und durch die Replikation ausfallsicher. Dadurch wird die Informationssicherheit als „vollständig“ bewertet (vgl. Tabelle 2.5).

Die PPB verzichtet mit dem eingeschränkten Zugriff auf die Blockchain und durch ausgewählte Nodes für den Konsens zunächst auf Dezentralität. Jedoch kann dies durch die Unterteilung in Admin, aktive und passive Node-Gruppen wieder erlangt werden. In den Node-Gruppen teilt sich die Verantwortung und die Arbeit. Gleichzeitig erlangen die Nodes eine Abhängigkeit durch die anderen. Aus diesem Grund lassen sich dezentrale Eigenschaften feststellen, die als „mittel“ zu bewerten sind (vgl. Tabelle 2.5).

Zentrale Datenbank

Im Rahmen dieser Arbeit wurden die Argumente der zentralen Datenbank für die Kriterien wie folgt dargelegt: Die Skalierbarkeit in einer zentralen Architektur ist stark eingeschränkt. Die gängigsten Mittel, die eingesetzt werden, um die Skalierbarkeit der zentralen Architektur zu erhöhen, machen diese zu einem verteilten System. Durch das Einsetzen eines Datenbankmanagementsystem (DBMS) können die vielen Anfragen, über eine Schnittstelle zur Datenbank, abgefangen und zusammengeführt werden. Um die Transaktionsgeschwindigkeiten des Servers zu erhöhen, besteht die Möglichkeit, die physischen Ressourcen eines Datenbankservers aufzustocken. Dadurch entsteht der Vorteil eines einfach erweiterbaren Systems, welches mit geringem administrativem Aufwand verwaltet werden kann. Eine zentrale Datenbank bewältigt damit gut die Aufgaben für Klein- bis Großunternehmen. Wenn es aber mit den anderen Technologien verglichen wird, mit denen eine internationale Größenordnung erreicht wird, ist die Skalierbarkeit als „mittel“ zu bewerten (vgl. Tabelle 2.5).

Hinsichtlich der Netzwerksicherheit verfügt eine zentrale Datenbank im lokalen Netzwerk über dieselben Angriffspunkte wie eine PPB. Es sollten nur autorisierte Zugänge gewährt werden. Nutzer des Datenbanksystems sollten darauf hingewiesen werden, dass dem Unternehmen, durch fahrlässiges Handeln hinsichtlich Softwaredownloads oder dem Öffnen nicht vertrauenswürdiger E-Mails, erheblicher Schaden zugefügt werden kann. Auch wenn eine zentrale Datenbank im Web angewendet wird, bietet eine standardisierte Architektur sowie gut geschriebener Code eine Verteidigung gegen SQL-Injection Angriffe. Zuletzt darf nicht die Aktualisierung der Infrastruktur fehlen. Hier entstehen die meisten Sicherheitslücken für das Datenbanksystem. Ein Datenbanksystem ist niemals sicher, aber einem zentralen Datenbanksystem stehen genügend Werkzeuge zur Verfügung, um das Datenbanksystems ausreichend zu schützen. Die eingesetzten Sicherheitsmaßnahmen sorgen für einen positiven Einfluss auf die Bewertung, weshalb die Netzwerksicherheit als „vollständig“ bewertet wird (vgl. Tabelle 2.5).

Die Informationssicherheit eines zentralen Datenbanksystems kann mit den 9 Codd’schen Regeln einfach umgesetzt werden. Damit ist sichergestellt, dass die Daten und Information dieses Datenbanksystems stets integer sind. Auf der anderen Seite bestehen hinsichtlich der Architektur eines zentralen Datenbanksystems Problematiken hinsichtlich der Verfügbarkeit und Ausfallsicherheit. Gibt es nur einen physikalischen Server mit allen Daten und der Server benötigt

neue Updates oder Erweiterungen, ist der Server und das darauf liegende Datenbanksystem in der Zeit des Updates nicht verfügbar. Hierdurch müssen in dem Fall Arbeiten an dem Datenbanksystem pausiert werden, was einem Ausfall des Systems ähnelt, nur dass dies vorher kommuniziert wird (vgl. Kapitel 4.2.3.). Um die Vertrauenswürdigkeit der Informationen zu gewährleisten, wird pro Client ein verschlüsselter Kommunikationskanal eingerichtet. Dies beeinflusst die Kriterien der Informationssicherheit positiv und wird damit als „gut“ bewertet (vgl. Tabelle 2.5).

In einem zentralen Datenbanksystem ist das Kriterium der Dezentralität nur insofern umsetzbar, als es durch ein DBMS möglich ist, die Arbeit in geringem Maße aufzuteilen. Diese Ansätze sind zwar verteilt, aber nicht völlig dezentral. Dadurch bleibt die Zentralität des Datenbanksystems und die damit einhergehenden Eigenschaften erhalten. Die Dezentralisierung des Datenbanksystems ist also als „nicht vorhanden“ bewertet (vgl. Tabelle 2.5).

Verteiltes Datenbanksystem

Bei der Einordnung anhand der Kriterien des verteilten Datenbanksystems, führen die Argumente der Skalierbarkeit zu dem Ergebnis, dass diese Architektur gut skaliert, wenn die erforderlichen Methoden eingesetzt werden. Dazu gehört zunächst die Erweiterung von der Client/Server-Architektur zu einer Multi-Client/Multi-Server-Architektur. Dadurch ist ein Scale-up möglich, womit die Serverressourcen entsprechend der Anforderungen der Clients mitwachsen. Ist ein Unternehmen an mehreren Standorten vertreten, werden die Daten durch die verteilte Architektur auf andere Server verteilt. Um dabei nicht alle Daten redundant zu speichern, werden diese fragmentiert. Nichtsdestotrotz muss berücksichtigt werden, dass die Skalierbarkeit mit steigender Anzahl der Standorte und dadurch auch dem erhöhten Verwaltungsaufwand abnimmt. In der Bewertung führt dies zwar zu einem weniger positiven Ergebnis, welches aber dennoch als „gut“ ausfällt (vgl. Tabelle 2.5).

Dadurch, dass eine verteilte Architektur eine Weiterentwicklung einer zentralen Architektur ist, bleiben die Argumente für eine gute Netzwerksicherheit gültig (vgl. Kapitel 4.1.2 – Netzwerksicherheit zentrale Datenbanksysteme). Bei einem verteilten Datenbanksystem ist eine erhöhte Vertrauenswürdigkeit für die interne Unternehmenskommunikation notwendig. Dies gelingt mit einem Session-Key, womit auch die Authentifizierung mit der notwendigen Integrität gewährleistet ist (vgl. 4.3.2 Netzwerksicherheit verteiltes Datenbanksystem). Eine vollständige

Netzwerksicherheit ist in einem verteilten Datenbanksystem schwieriger zu erreichen, da es zusätzliche Angriffspunkte gibt (vgl. Kapitel 4.3.2). Daraus resultiert das Ergebnis „gut“ (vgl. Tabelle 2.5).

Bei der Informationssicherheit in verteilten Datenbanksystemen müssen die Daten und Informationen auf allen Replikationen stets integer und vollständig sein, weshalb ein größerer Aufwand für die Verwaltung und die Synchronisation notwendig ist. Jedoch sorgen die zusätzlichen Standorte dafür, dass das Unternehmensnetzwerk und die Services ausfallsicherer sind. Sie sorgen außerdem für eine höhere Verfügbarkeit der Ressourcen. Sollte Ressource A an Standort D durch eine Sperre nicht verfügbar sein, kann die Anfrage an eine andere Datenbank an Standort E weitergeleitet werden. Wird dabei auf eine sichere Kommunikation durch einen Session-Key geachtet, ist die Informationssicherheit für ein verteiltes Datenbanksystem als „vollständig“ zu bewerten (vgl. Tabelle 2.5).

Ein verteiltes Datenbanksystem versucht nach außen dezentral zu wirken. Ein Service wirkt für den Kunden so, als ob dieser auf seinem Rechner durchgeführt wird. Dabei sendet das Frontend die Daten an die Middleware, die dann dafür sorgt, dass die Aufgabe von dem korrekten Server bearbeitet wird. Die lokale Unabhängigkeit der verschiedenen Datenbanken trägt zu einer positiven Bewertung bei. So erfüllt ein verteiltes Datenbanksystem nur sinngemäß die Kriterien der Dezentralität, sodass es als „gering“ bewertet wird (vgl. Tabelle 2.5).

Cloud Datenbank

Die Ergebnisse aus der Einordnung der Clouddatenbank durch die Kriterien lassen sich wie folgt zusammenfassen: Bei einer Clouddatenbank ist die Skalierbarkeit sehr flexibel und hängt von den finanziellen Ressourcen ab. Der Cloud Provider bietet vieles an, angefangen bei virtuellen Server, bis hin zu einer kompletten Applikation, die eine Clouddatenbank nutzt. So können für ein spontan erhöhtes Aufkommen zusätzliche Ressourcen gebucht und wieder abbestellt werden, womit die Skalierbarkeit als „vollständig“ zu betrachten ist (vgl. Tabelle 2.5).

Bei der Netzwerksicherheit wurde aufgezeigt, dass eine Cloud-Architektur und die entsprechende Datenbank von dem Zugriff durch das Internet abhängig sind. Wird der Service durch einen DDoS-Angriff beeinträchtigt, so wird nicht nur dem Nutzer, sondern auch dem Betreiber geschadet. Mittel, wie eine Filterung von IP, sind für eine öffentliche Cloud schwieriger

umzusetzen als für ein privates Netzwerk, bei dem mit White-Listen gearbeitet wird. Durch den erhöhten Einsatz der Cloud-Lösungen von Unternehmen im Jahr 2021, haben sich die Cloud-Provider zu beliebten Zielen für Angriffe gemacht. Dies wird daraus geschlussfolgert, dass jährlich mehr Unternehmen eine Cloud-Lösung einsetzen (siehe Abs. 4.4.2). Damit wird die Netzwerksicherheit für ein Cloud-Datenbanksystem als „Mittel“ bewertet.

Der Abschnitt der Informationssicherheit hat erwiesen, dass der Fokus eines Cloud-Datenbanksystems darin liegt, einen sicheren Kommunikationskanal vom Provider zum Nutzer zu garantieren. Der sichere Kommunikationskanal wird durch eine SSL-Verschlüsselung ermöglicht, die sich außerdem für die Vertrauenswürdigkeit verantwortlich zeigt. Durch zusätzliche digitale Zertifikate und Prüfsummen kann die Integrität der Daten festgestellt werden. Eine Cloud-Architektur besitzt - aufgrund einer hohen Ausfallsicherheit durch die redundante Infrastruktur und durch Fragmentierung auf einer Privaten-Cloud - die beste Verfügbarkeit. Damit ist die Informationssicherheit für Cloud-Datenbanksysteme als „vollständig“ zu bewerten (vgl. Tabelle 2.5).

Aufgrund des Definitionsunterschieds von verteilt und dezentral (Abs. 3.6 Dezentralisierung) weisen die Argumente bei einem Clouddatenbanksystem daraufhin, dass die Dezentralisierung „nicht vorhanden“ ist. Die Verantwortung und Weiterentwicklung eines Cloud-Datenbanksystems werden durch den entsprechenden Provider getragen. Ein Cloud-Datenbanksystem wirkt, wie ein kohärentes System mit dezentralen Eigenschaften, dies sind aber keine dezentrale Eigenschaften, sondern verteilte.

Tabellarische Zusammenfassungen

Im folgenden Abschnitt werden die zuvor vorgestellten Ergebnisse tabellarisch zusammengefasst und anhand der Bewertungstabelle (vgl. Tabelle 2.5) mit Punkten bewertet. Diese Punkte werden am Schluss durch die Anzahl der Subkriterien geteilt und erhalten eine Durchschnittspunktzahl pro Kriterium. Es folgen Tabellen zur Skalierbarkeit, Netzwerksicherheit,

Informationssicherheit und Dezentralität. Die Skalierbarkeit wird in die folgenden Subkriterien differenziert: Größen-Skalierbarkeit, Geographische Skalierbarkeit und Administrative Skalierbarkeit. Bei der Netzwerksicherheit wird die Tabelle so aufgeteilt, wie sich die Technologien gegen SQL-Injection-Angriff, DDoS-Angriff, unautorisierten Zugriff und Advanced Persistent Threat verteidigen können. Dabei wird bewertet, wie viele und wie effektiv die Möglichkeiten sind (0-4 Punkte). Bei der Informationssicherheit ist die Tabelle in Verfügbarkeit, Vertrauenswürdigkeit und Integrität unterteilt. Dafür werden die Punkte so vergeben, wie erfolgreich diese Eigenschaften umgesetzt werden können. Zuletzt wird bewertet, inwiefern eine Dezentralität in den Technologien umsetzbar ist. Für die Erläuterung wird sich lediglich auf die hier erwähnten und interpretierten Argumente dieser Arbeit bezogen.

	Größen-Skalierbarkeit [Bewertung – Erläuterung]	**Geographische Skalierbarkeit** [Bewertung – Erläuterung]	**Administrative Skalierbarkeit** [Bewertung – Erläuterung]	**Ergebnis** **[**Punkte/Anzahl der Kriterien]
PPB	4 - Grundlegende eingesetzte Methoden sorgen für eine gute Skalierbarkeit in der Kommunikation, physische Ressourcen wachsen dynamisch mit	3 - Abhängig vom Standort der Unternehmen mit denen eine Blockchain zusammen benutzt wird	2 - Hoher Aufwand durch die Abstimmung der unterschiedlichen Interessen der Unternehmer lassen sich aber gut durch Smart-Contracts festhalten	9/3 = 3
Zentrales Datenbanksystem	2 - Physikalische Ressourcen können einfach aufgerüstet werden allerdings nur solange es die Infrastruktur zulässt	0 - Nicht vorhanden, da meist im LAN gearbeitet wird	4 - Geringerer Aufwand durch einen zentralen Standort	6/3 = 2
Verteiltes Datenbanksystem	4 – Verteilte Systeme erhöhen die Skalierbarkeit der Kommunikation, Ressourcen und Aufgaben können gut verteilt werden	4 – Durch die Vernetzung von mehreren Standorten liegt die Verantwortung der Ressourcen dennoch beim Unternehmen	2 - Höherer Aufwand durch verschiedene Standorte und lokaler Unabhängigkeit	10/3 = 3,5
Cloud Datenbanksystem	4 – Hohe Kapazitäten durch eine Serverfarm	4 - Überall aus der Welt nutzbar, bei einem Zugang zum Internet	2 - Hoher Aufwand durch zentralen Standort mit hohen Kapazitäten	10/3 = 3,5

Tabelle 2.6 – Ergebnisse Skalierbarkeit

	SQL – Injection Angriff (siehe Abs. 4.2.2) [Bewertung – Erläuterung]	**DDoS – Angriff (siehe Abs. 4.1.2 und 4.4.2)** [Bewertung – Erläuterung]	**Unautorisierter Zugriff (siehe Abs. 4.1.2)** [Bewertung – Erläuterung]	**Advanced Persistent Threat (siehe Abs. 4.1.2)** [Bewertung – Erläuterung]	**Ergebnis** [Punkte/Anzahl der Kriterien]
PPB	4 – Wird meist mit NoSQL Datenbanken benutzt	3 – Gut verhinderbar durch gezielte Filterung und Whitelists	4 – Durch digitale Signaturen und einem Access Management vollständig kontrollierbar	2 – Durch mangelhaften Code und schlechten Smart-Contracts können Lücken entstehen	13/4 = 3,25
Zentrales Datenbanksystem	3 – Durch guten Code verhinderbar, aber nicht vollständig	4 – Infrastruktur ermöglicht eine gute Verhinderung abhängig davon, ob es eine Web-Datenbank ist	4 – Vollständig kontrollierbar durch zentrale Domäne und eigenem Access Management.	2 – Risikofaktor durch Malware per Mail oder veraltete Serversysteme	13/4 = 3,25
Verteiltes Datenbanksystem	3 - Durch guten Code verhinderbar, aber nicht vollständig	2 – Mehr Schnittstellen für einen möglichen Angriff durch verteilte Kommunikation.	3 – Gut kontrollierbar durch ein Access Management jedoch durch die vielen verschiedenen Standorte nicht immer sofort vertrauenswürdig	2 – Erhöhter Risikofaktor durch die Masse an möglichen Nutzern und die Nichteinhaltung von Sicherheitsmaßnahmen.	10/4 = 2,5
Cloud Datenbanksystem	3 - Durch guten Code verhinderbar, aber nicht vollständig	2 – Öffentlicher Zugang ermöglicht einen weltweiten Angriff	2 – Cloud-Nutzer registrierten über 50 % unautorisierten Zugang trotz einer Proxy-Lösung vom Provider	3 – Durch die hohen Kapazitäten gibt es Replikationen der genutzten Server, weshalb korrumpierte Systeme einfacher ausgeschaltet werden können	10/4 = 2,5

Tabelle 2.7 – Ergebnisse Netzwerksicherheit

	Verfügbarkeit [Bewertung – Erläuterung]	**Integrität** [Bewertung – Erläuterung]	**Vertrauenswürdigkeit** [Bewertung – Erläuterung]	**Ergebnis** [Punkte/Anzahl der Kriterien]
PPB	3 – Durch ein redundantes Distributed Ledger und einem Konsensmechanismus können Transaktionen in einer Queue abgearbeitet werden, allerdings sind bei einer Off-Chain Lösung die Informationen lokal abhängig	4 – Durch das Loggen von Transaktionen ist immer nachvollziehbar wann Daten/Informationen verändert wurden	4 – Durch Private/Public Keys und digitalen Signaturen, die durch Smart-Contracts geprüft werden	11/3 = 3,66
Zentrales Datenbanksystem	3 – Meist eingeschränkt durch eine Schnittstelle, wird durch ein DBMS jedoch aufgefangen	4 - Durch Einhalten der Codd'schen Regeln werden Informationen integer gehalten	4 – End-zu-End Verschlüsselung und sichere Kommunikationskanäle	11/3 = 3,66
Verteiltes Datenbank-system	4 – Durch Fragmentierung und Replizierte Datenbanken kann eine Anfrage durch das DDBMS stets bearbeitetet werden	3 – Erhöhter Aufwand Informationen Konsistent zu halten durch die viele Verteilung	4 – Mittels Session-Keys vollständig vertrauenswürdig	11/3 = 3,66
Cloud Datenbanksystem	4 – Durch hohe Kapazitäten und Fragmentierung	4 – Daten und Informationen werden nur durch eine Datenbank repräsentiert und durch Prüfsummen gesichert	4 – Mittels SSL und Digitalen Signaturen	12/3 = 4

Tabelle 2.8 – Ergebnisse Informationssicherheit

	Umsetzbar [Bewertung – Erläuterung]	**Ergebnis** [Punkte/Anzahl der Kriterien]
PPB	2 – Durch P2P können alle Nodes dasselbe und durch entsprechende Node-Gruppen wird die Arbeit unter sich aufgeteilt, jedoch darf niemals allein entschieden werden	2/1 = 2
Zentrales Datenbanksystem	0 – Aufgrund der zentralen Einheit ist eine Verteilung der Aufgaben nicht vorgesehen und mit dem DBMS werden nur verteilte Ansätze geschaffen	0/1 = 0
Verteiltes Datenbanksystem	1 – Durch lokale Unabhängigkeit können Aufgaben selbstständig getätigt werden, allerdings bedarf es eine Synchronisation aller Datenbanken	1/1 = 1
Cloud Datenbanksystem	1 – System ist strikt in Client/Server-Architektur geteilt, jedoch können weitere Datenzentren lokal unabhängig arbeiten	1/1 = 1

Tabelle 2.9 - Ergebnisse Dezentralität

5.2 Diskussion der Erkenntnisse

Um aufzuzeigen, wofür eine PPB besser geeignet ist als zentrale, verteilte und Cloud-Datenbanksysteme, wurden eigene Kriterien auf Basis von bestehender wissenschaftlicher Literatur definiert. Für die Einordnung und Bewertung der Datenbanklösungen wurden anhand der Kriterien Argumente zusammengetragen, wodurch eine Vergleichbarkeit der Ergebnisse hergestellt werden konnte.

Aus den Tabellen ist zu entnehmen, dass die PPB in der Skalierbarkeit, Netzwerksicherheit, Informationssicherheit und der Dezentralität eine durchschnittliche Bewertung von 3 Punkten erhält. Damit ist diese Technologie in der Lage, die definierten Kriterien „gut“ zu erfüllen. In allen Aspekten ist eine PPB positiver bewertet als ein zentrales Datenbanksystem. Die Kriterien für die Skalierbarkeit wurden derart gewählt, um die Eigenschaften eines gut verteilten Systems

widerzuspiegeln. Was nicht in die Bewertung der Skalierbarkeit für eine zentrale Datenbank geflossen ist, ist die einfache Weiterentwicklung zu einer verteilten Datenbank. Dies hat zur Folge, dass die Skalierbarkeit eines zentralen Datenbanksystems als „mittel“ bewertet wird. Im Aspekt der Skalierbarkeit weisen die Cloud-Datenbanksysteme und verteilte Datenbanksysteme in der Geographischen Skalierbarkeit eine bessere Bewertung auf als die PPB. Dies hat die Ursache, dass in der PPB alle genutzten Ressourcen in der eigenen Verantwortung liegen. Man ist von der Verfügbarkeit aktiv teilnehmenden Nodes anderer Unternehmen abhängig.

Die Ergebnisse zeigen bezüglich der Netzwerksicherheit, dass die PPB mit zentralen Datenbanksystemen am besten bewertet wurde. Dies ist deshalb der Fall, weil in einer PPB nur Nodes eine SQL-Anfrage an andere Nodes schicken. Dabei prüfen die Nodes, ob die Anfrage gültig ist, entspricht die Anfrage nicht dem vereinbarten Standard, wird diese abgewiesen. In zentralen Datenbanksystemen ist die bessere Netzwerksicherheit dadurch möglich, dass in einem nicht öffentlichen Netzwerk besser gefiltert werden kann. Somit sind die internen Unternehmensservices nicht so anfällig für DDoS-Angriffe.

Möglicherweise liegt dies daran, dass in dem Netzwerk keine händischen Transaktionen getätigt werden, sondern im Optimalfall nur automatische Transaktionen. Das PPB-Netzwerk stellt den Service zum Austausch von Daten zur Verfügung. Wenn ein Netzwerk ausschließlich dafür verwendet wird, ist das Netzwerk automatisch restriktiver. Zusätzlich ist jederzeit nachvollziehbar, wer Transaktionen getätigt hat und wann welche getätigt wurden. Dadurch wird ein Missbrauchen des Netzwerks vermindert und ein Debugging ermöglicht, sollte das Netzwerk kompromittiert sein. Hinsichtlich der Argumente für die Informationssicherheit sind die Technologien gleichgut bewertet, mit der Ausnahme, dass die Clouddatenbank eine bessere Verfügbarkeit aufweist. Eine Clouddatenbank stellt eine enorme Menge an Ressourcen zur Verfügung, weshalb die Clouddatenbank in der Informationssicherheit besser bewertet wurde als der Rest.

Zudem haben die Ergebnisse der Dezentralität gezeigt, dass die PPB die einzige Technologie ist, die eine echte Dezentralität aufweisen kann. In einem Blockchain-Netzwerk sind neben den unterschiedlichen Berechtigungsgruppen die Nodes gleich. Der Grundgedanke einer Dezentralität ist eine gleichberechtigte Aufgabenverteilung, sodass Entscheidungen nicht von einer Instanz getroffen werden, sondern zum Beispiel von einem Konsortium. Dieses faire und

nachvollziehbare Verhalten ist durch einen Konsensmechanismus gesichert. Das Verhalten ist stets deterministisch und erfolgt nach Richtlinien, die die teilnehmenden Admin-Nodes über Smart-Contracts definiert haben. Ebenso ist die Datenbank der Blockchain dezentral, denn auf jedem Node ist eine eigene Kopie des Distributed Ledger enthalten. Nicht zu vergessen ist die Tatsache, dass bei einem Cloud-Datenbanksystem und einem verteilten Datenbanksystem hinsichtlich der Dezentralität die Datenbanksysteme keinen höheren Nutzen erhalten. Denn bei einer vollen Dezentralität hieße dies, dass auf jedem Client die gesamte Datenbank läge, was zu einer erheblichen Ressourcenverschwendung führen würde. Solch ein hohes Maß an Dezentralität wäre nur dann möglich, wenn die Information, die geteilt würde, wenige MB groß wäre. Sobald man anfangen würde, nicht strukturierte Daten abzuspeichern, wäre die Ressourcenkapazität auf den Endgeräten schnell erreicht.

Eine weitere Erkenntnis aus dem Vergleich der Ergebnisse aus den Tabellen 2.6 und 2.7 ist, dass eine erhöhte Skalierbarkeit mit einer schlechteren Netzwerksicherheit einhergeht, oder dass eine schlechtere Skalierbarkeit mit einer besseren Netzwerksicherheit möglich ist. Bei einem zentralen Datenbanksystem ist die Skalierbarkeit sehr beschränkt, dafür lassen sich in der Netzwerksicherheit viele Methoden anwenden, die schnell und ohne große Abstimmung durchgeführt werden können. Anders ist dies bei dem Rest, in einem Cloud-Datenbanksystem ist die Netzwerksicherheit durch die öffentliche Verfügbarkeit ständigen Angriffen ausgesetzt. Das verteilte Datenbanksystem erlangt eine höhere Skalierbarkeit durch die zusätzlichen Server-Standorte, dies hat jedoch den Nachteil, dass es dadurch mehrere Zugänge ins Unternehmensnetzwerk gibt. Bei der PPB wächst die Skalierbarkeit mit jedem Node und den teilnehmenden Unternehmen, welche dem Netzwerk beitreten. Damit müssen neuen und fremden Nodes Zugang gewährt werden, die ohne zusätzliche Prüfung ein Risiko für das Netzwerk sind.

Abschließend ist zu sagen, dass sich dieser Vergleich auf zentrale, verteilte und Cloud Datenbanken und der PPB beschränkt. Für eine ausführlichere Forschung sollte die PPB mit anderen Blockchain-Lösungen oder ähnlichen Netzwerkstrukturen verglichen werden, die es ermöglichen, Daten und Informationen sicher und transparent auszutauschen.

5.3 Reflexion der genutzten Literatur

In diesem Abschnitt soll näher aufgeführt werden, warum gewisse Quellen verwendet wurden und weshalb diese zu hinterfragen sind. Die erste Quelle ist ein wissenschaftliches Kapitel mit dem Titel: „Data Storage in a Decentralized World: Blockchain and Derivatives“ von Enis Karaarslan und Enis Konacakli aus dem Buch „Who Runs the World: Data“ von Sevinç Gülseçen, Sushil Kumar Sharma, Emre Akadal aus dem Jahr 2020 von der Universität Istanbul. Diese Quelle war eine der ersten, die für diese Arbeit verwendet wurde. In Kapitel 5 der genutzten Quelle, wird die Blockchain mit zentralen Datenbanken verglichen. Dies erbrachte die grundlegende Idee zur Erforschung der Blockchain-Technologie im Vergleich mit anderen Datenbanken. Die Kritik an dem Vergleich aus der Quelle ist, dass die Blockchain selbst keine Datenbank ist. Es wird zwar zwischen relationalen Datenbanken und nicht-relationalen Datenbanken unterschieden, um dann die zentrale Datenbank als relational und die Blockchain als nicht-relational anzusehen. In dieser Betrachtung wird allerdings zwischen einer Datenbank und einem Datenbanksystem unterschieden. Die Blockchain installiert per Default eine Datenbank, welche in dieser Arbeit als Distributed-Ledger vorgestellt wurde. Daher nutzt die Blockchain-Technologie eine Datenbank, statt eine zu sein. Anhand der gesammelten Erkenntnisse aus dieser Arbeit ist das Blockchain-Datenbanksystem ein ganzes Netzwerk. Dieses Netzwerk ist durch Peers verbunden und ermöglicht den Austausch von Daten zwischen den Peers, weshalb eine Blockchain eher mit anderen Architekturen verglichen werden sollte, wie z. B. ein verteiltes Datenbanksystem (Distributed Systems) oder einem Clouddatenbanksystem.

Ähnliche Gedanken zu einem Use Case einer Permissioned Blockchain wurden in dem Konferenzpapier von (Xiao, et al., 2018) vorgestellt, hierbei handelt sich um eine Public Permissioned Blockchain. Dieses Papier hat einen praktischen Nutzen einer Off-Chain aufgezeigt und grundlegende Inspiration für den Abschnitt 5.2 zu einer praktischen Implikation geliefert. In der wissenschaftlichen Arbeit wird vorgestellt, wie eine Software in der Praxis aussehen könnte, damit verschiedene Organisationen ihre medizinischen Daten austauschen können. Kritisch an diesem Papier ist, dass es „data owner“ gibt, welche Anderen anonymisierte Patientendaten zur Verfügung stellen. Nach der DSGVO §17.1 haben Menschen der EU-Mitgliedstaaten das Recht auf Löschung der personenbezogenen Daten. Nach Artikel §4.1 sind personenbezogene Daten auch dann personenbezogen, wenn die Person „[…] zu einem oder

mehreren besonderen Merkmalen, die Ausdruck der physischen, physiologischen, genetischen, psychischen […] Identität dieser natürlichen Person sind, identifiziert werden kann.“[127]. Das bedeutet auch bei Pseudonymisierung oder Anonymisierung des Patienten könnten Ärzte den Befund noch immer einer Person zu ordnen, wenn z. B. die Krankheit sehr selten ist. Das Recht auf Löschung der Daten steht auf dem ersten Blick mit den Distributed Ledger im Konflikt. Dort soll unwiderruflich festgehalten werden, dass die Daten von Patient X an Arzt Y transferiert werden. Jedoch wäre es, ohne das Distributed Ledger nicht möglich nachzuweisen, wohin die Daten transferiert wurden. Damit wurde mit dem Konferenzpapier die Erkenntnis gewonnen, dass man von einer praktischen Umsetzung nicht mehr weit entfernt ist. Allerdings muss in Europa noch zusätzlich in Bezug auf Datenschutz geforscht werden.

Gesamtheitlich bedeutet es, dass die Blockchain-Technologie noch unzureichend analysiert ist und es weiteren Bedarf gibt, die Forschungen zu unterstützen.

5.4 Praktische Handlungsempfehlung

In diesem Abschnitt sollen die Erkenntnisse aus dieser Arbeit für eine praktische Implikation erneut aufgegriffen werden. Dafür wird die erste und dritte Forschungsfrage ausführlicher beantwortet. Zu Beginn die erste Frage, die wie folgt gestellt wurde: „Inwiefern ist die Blockchain-Zugriffsbeschränkung „Private Permissioned“ für ein Unternehmen geeignet?“.

5.4.1 Was bedeutet Private Permissioned für die Blockchain?

Beim Verarbeiten der Quellen von (Thießen, 2020), (Pohlmann, 2019) und (Schurtenberger, 2022) bezüglich Private-Blockchain, gab es meist eine Gegenüberstellung mit Public-Blockchain. Diese Gegenüberstellung zeigt, dass die Public-Blockchain von viel mehr Nutzern genutzt werden kann, da sie öffentlich zugänglich ist. Damit das möglich ist, wird die Technologie als Open-Source-Projekt angesiedelt. Somit können alle den Code lesen und Funktionen

[127] Art. 4 DSGVO – Begriffsbestimmungen - Datenschutz-Grundverordnung (DSGVO) (dsgvo-gesetz.de)

testen. Neue Funktionen sind schneller ausprobiert und Sicherheitslücken fallen schneller auf, da mehr Menschen den Code sehen.[128] Durch das Privatisieren einer Blockchain gehen die Open-Source Vorteile verloren. Wird die Blockchain in einem oder für ein Unternehmen entwickelt, liegt das Fachwissen bei denen die sie entwickeln.

Daraus folgt, dass bei neu entstehenden Problemen oder neuen Funktionen die Blockchain von den jeweiligen Fachkräften weiterentwickelt werden muss. Eine Folge davon ist, dass Features oder Bugs, je nach Ressourcen, langsamer entwickelt oder gelöst werden.

Neben der Administration der Blockchain, bedarf es noch einer zusätzlichen Instanz, die die Ledger-Ebene der Blockchain verwaltet und festlegt, wer neue Blöcke zur Blockchain hinzufügen und validieren darf. Diese zugelassenen Nodes wenden je nach Konsensmechanismus eigene Ressourcen auf. Daraus resultiert, dass für eine Validierung stetig Energie und Ressourcen von allen Nodes eingesetzt werden muss, um jede Transaktion an ihr Ziel zu bringen.

5.4.2 Welche Probleme und Chancen ergeben sich für ein Unternehmen beim Einsetzen einer PPB?

Die Grundfunktion einer Blockchain ist zunächst eine durch Kryptografie gesicherte Übertragung der Informationen. Um als Unternehmen Informationen intern untereinander auszutauschen, bedarf es keiner Blockchain. Eine Blockchain-Lösung ergibt dann aber Sinn, wenn Informationen mit nicht vertrauenswürdigen Partnern ausgetauscht werden sollen.[129] Dafür muss ein Unternehmen Kooperationen finden, die schon ein Blockchain-Netzwerk nutzen oder bereit wären, eines zu entwickeln.

Wenn akzeptiert wird, dass das Einsetzen und Entwickeln einer Blockchain Geld und Zeit kosten wird können sich daraus neue Chancen ergeben. Eine davon ist die Möglichkeit

[128] (Schurtenberger, 2022)

[129] vgl. (Karaarslan & Konacaklı, 2020)

Informationen und Daten mit nicht vertrauenswürdigen Partnern, durch ein vollständig kontrolliertes Netzwerk auszutauschen. [130]

Einer der ersten Use Cases dazu war eine Lieferkette. In einer Lieferkette sind viele verschiedene Organisationen involviert. Ein Problem an nicht digitalen Lieferketten ist, dass die Herkunft der einzelnen Teile eines Produktes für die Endkunden nicht nachvollziehbar ist. Diese Information kann nur durch den Hersteller selbst zur Verfügung gestellt werden. Also bedarf es einer Plattform, die transparent, aber auch sicher ist. Dieses Problem könnte eine PPB mit einer Hyper-Ledger-Plattform lösen.[131] Demzufolge könnte ein Unternehmen ein höheres Vertrauen zu seinen Kunden aufbauen, indem der Kunde den Lieferprozess durch transparente Transaktionen nachvollziehen kann.

5.4.3 **Medical Health System mit einer PPB**

Die dritte Forschungsfrage lautet: „Wofür ist die PPB hinsichtlich der diskutierten Kriterien besser geeignet als traditionelle Datenbanksysteme?“. Ein Szenario ergibt sich in der Medizin. Es gibt Kliniken, die ihre Daten noch immer physisch miteinander austauschen. Wenn beispielsweise eine Ultraschalluntersuchung gemacht wird, sind die Ultraschallbilder entweder ausgedruckt oder auf einer CD gebrannt. Ein Grund dafür könnte das Misstrauen in neue Technologien und deren Möglichkeiten sein. Oft sind die Kosten für Innovationsprojekte hoch, aber der reale Nutzen nicht greifbar. Dafür wurde 2015 im Deutschen Bundestag ein Bericht mit der Fragestellung „Der hiermit vorgelegt Innovationsreport des TAB fasst den wissenschaftlichen Kenntnisstand zu den Wirkungszusammenhängen von medizinisch-technischem Fortschritt und Gesundheitsausgaben, Wachstum und Beschäftigung sowie Gesundheitszustand der Bevölkerung zusammen“ veröffentlicht. Darin wurde festgestellt, dass bisherige technische Innovationen maßgeblich zu einer verbesserten Gesundheitsvorsorge und einer erhöhten Lebenserwartung beigetragen haben. Zudem hat die Weiterentwicklung im Bereich Prävention, Diagnose und Therapie, dazu beigetragen, dass Bürger und Bürgerinnen wieder aktiv am

[130] vgl. (Pohlmann, 2019)

[131] vgl. (Banerjee, 2018)

Gesellschaftlichen Leben teilhaben können.[132] Daraus lässt sich schließen, dass es eigentlich kein Misstrauen gegenüber Innovationen geben sollte.

Die PPB-Technologie soll mit ihrer Möglichkeit zum Austausch von Informationen einen Beitrag zu einer verbesserten Gesundheitsvorsorge leisten. Damit wird die Blockchain-Technologie wahrscheinlich nicht selbst zu einer Verbesserung sorgen können, allerdings ermöglicht das Einsetzen einer Blockchain einen gesicherten und transparenten Kommunikationskanal. Wird dazu ein Smart Contract vereinbart, so können die Daten für die Forschung genutzt werden.

Die Problematik an medizinischen Daten ist die Größe, vor allem bildgebende Daten wie DICOM. Dies ist ein gängiges Format für Untersuchungen wie MRT's oder Ultraschallbilder. Das Besondere an diesen Bilddaten ist, dass in jedem einzelnen Bild ein Informationsheader zu finden ist.[133] Die Größe der Daten variiert stark von der Anzahl an Bildern, dabei kann je nach Auflösung mit ca. 10-25 MB pro Bild gerechnet werden. Besteht z. B. eine vollständige MRT-Untersuchung aus 100 Bildern, so kann ein vollständiger Datensatz 1-2,5 GB groß sein. Mit dieser Größenordnung sollten Bilder nicht über eine Blockchain transferiert werden. Eine geeignetere Lösung wäre eine Off-Chain, wie sie im Abschnitt 4.1.1 vorgestellt wurde. Dabei existiert, zu dem Distributed Ledger, in dem die Transaktionen festgehalten wird, weitere Datenbanksysteme auf denen die Bilddaten abgelegt werden können. Wird ein reales Beispiel hinzugezogen, so könnte in einer Studie im Krankenhaus eine Vielzahl von Untersuchungen durchgeführt werden, in denen die bildgebende Daten entstehen. Für die Qualitätssicherung der Daten müssen die Studienleiter externes Fachpersonal beauftragen. Hierfür müssen die Daten sicher an das jeweilige Personal übertragen werden können. Die Qualitätsgesicherten Daten müssen danach wieder zurück in die Studie.

An dieser Stelle könnte die PPB-Lösung helfen. Mit der On-Chain-Lösung werden die Smart Contracts und jede Transaktion festgehalten. Dabei ist nachvollziehbar, wer die Datei verändert hat oder die Daten noch ihren Rohzustand hält. Für die eigentliche Datenübertragung wäre die

[132] vgl. (Lips, et al., 2015)

[133] Digital Imaging and Communications in Medicine, Overview (dicomstandard.org)

Blockchain nicht mehr in der Lage und es müsste mit einem Dateisystem gekoppelt werden. Hier würde die Off-Chain-Lösung ansetzen und eine direkte P2P-Verbindung der Smart Contract-Partner aufbauen, über diesen dann die größeren Daten übertragen werden können.[134]

[134] vgl. (Xiao, et al., 2018)

6 Fazit

6.1 Wurde das Ziel erreicht?

Indem sich in diese Arbeit vor allem mit der PPB befasst hat, war ein Vergleich mit traditionellen Datenbanksystemen erst möglich. Anfänglich war es kritisch, nur mit dem Begriff Blockchain zu arbeiten, da dieser nur eine Technologie beschreibt, in der verkettete Datenblöcke mit Informations-Header verwaltet werden. Aufgrund des in der Blockchain fehlenden DBMS war es schwierig die Blockchain als ein Datenbanksystem zu betrachten. Um die Vergleichbarkeit der Technologien zu gewährleisten, hat sich diese Arbeit auf eine PPB fokussiert.

Die PPB wurde auf der konzeptionellen Ebene mit zentralen, verteilten und Cloud-Datenbanksysteme verglichen. Ziel dieser Arbeit war es dann, die Technologien vorzustellen und anhand von definierten Kriterien: Skalierbarkeit, Netzwerksicherheit, Informationssicherheit und der Dezentralität zu bewerten. Die Resultate in dieser tabellarischen Bewertung haben gezeigt, dass die PPB ähnliche, sowie bessere Ergebnisse geliefert hat als die traditionellen Datenbanksysteme. Dies weist darauf hin, dass eine Blockchain-Lösung in Unternehmen einen zukunftsfähigen Bestand haben kann. Vor allem ist die Blockchain-Technologie in der Lage, das Problem zu lösen, Daten und Informationen mit Organisationen oder Unternehmen auszutauschen, denen nicht vollends vertraut werden kann.

Aus der Ausarbeitung resultierte eine Handlungsempfehlung für das Medical Health System. Die Blockchain-Technologie bietet die Möglichkeit, dass eine Studie ihre erhobenen Forschungsdaten an nicht vertrauenswürdige Forscher senden kann. Durch Smart Contracts ist es möglich, dass rechtliche Regularien automatisch gesichert werden und mit dem Distributed Ledger der Blockchain werden alle beteiligten Personen und Datenblöcke, sowie die dazugehörigen Uhrzeiten jeder Transaktion geloggt.

Damit können die gewonnenen Erkenntnisse der Arbeit für weitere Forschungen verwendet werden. Sollten die Technologien in einen praktischen Zusammenhang gestellt werden, so können die Ergebnisse dieser Arbeit von den praktischen Resultaten abweichen.

6.2 Ausblick

Die PPB sollte mit weiteren konzeptionellen Technologien verglichen werden, gegebenenfalls mit einem praktischen Bezug, um eine bessere Einschätzung zu erhalten, wie die verschiedenen technischen Methoden zusammenarbeiten. Dafür kann die Transaktionsgeschwindigkeit der einzelnen Datenbanksysteme als zusätzliches Kriterium hinzugezogen werden.

Diese Arbeit hat darauf verzichtet sich auf einen Konsensmechanismus festzulegen, sondern die Kernidee geliefert, wie ein Konsens auszusehen hat. Bei der mittlerweile breiten Masse an verschiedenen Mechanismen, sollte erforscht werden welcher Konsensmechanismus sich für eine PPB besser eignet.

Literaturverzeichnis

A. Davenport, S. S. a. X. L., 2018. *Attack Surface Analysis of Permissioned Blockchain Platforms for Smart Cities.* Kansas City, IEEE Xplore, pp. 1-6.

Abadi, J. & Brunnermeier, M., 2022. *Blockchain Economics,* Cambridge: s.n.

Anita N, V. M., 2019. *Blockchain Security Attack: A Brief Survey.* s.l., s.n., pp. 1-6.

Anon., 2022. *Bundesamt für Sicherheit in der Informationstechnik.* [Online] Available at: https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Moderner-Staat/ElektronischeSignatur/Glossar/esigglossar.html?nn=450092#Glossar_I
[Zugriff am 24 Februar 2022].

Banerjee, A., 2018. *Blockchain Technology: Supply Chain Insights from ERP,* India: s.n.

Barbara, C., 2009. *Digital Signatures,* Boston: s.n.

Barkai, D., 2001. *Technologies for sharing and collaborating on the Net.* [Online] Available at: https://doi.org/10.1109/P2P.2001.990419
[Zugriff am Februar 2022].

Beutelspacher A, S. J. W. K.-D., 2015. *Moderne Verfahren der Kryptographie: Von RSA zu Zero-Knowledge.* Wiesbaden: Springer Spektrum.

Blackblaze, 2022. *Hard Drive Data and Stats,* s.l.: Blackblaze Homepage.

BMWK & Michael, H., 2022. *Self Sovereign Identity für Deutschland - Digitale Technologien.* [Online]
Available at: https://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/SchaufensterSichereDigIdentProjekte/sdi-projekt_ssi.html
[Zugriff am März 2022].

Bundesamt für Sicherheit in der Informationstechnik, 2020. *Relationale Datenbanksysteme,* Deutschland: Bundesamt für Sicherheit in der Informationstechnik.

Buterin, V., 2020. *Ethereum Wiki.* [Online] Available at: https://eth.wiki/ [Zugriff am 23 02 2022].

Clarke, J., 2009. *SQL Injection Attacks and Defense.* USA: Elsevier.

Connolly, T., Begg, C. & Strachan, A., 2002. *Datenbanksysteme.* München: Addison-Wesley Verlag.

Dadam, P., 2013. *Verteilte Datenbanken und Client/Server-Systeme: Grundlagen, Konzepte und Realisierungsformen.*. s.l.:Springer-Verlag.

Daniela Soares Cruzes, M. F. T. D. O. M. G. I. P., 2017. How is Security Testing Done in Agile Teams? A Cross-Case Analysis of Four Software Teams. In: *Agile Processes in Software Engineering and Extreme Programming.* Köln: Springer Open.

David, B., 2002. *Peer-to-Peer Computing.* [Online] [Zugriff am März 2022].

Demeester, T., 2017. *Critique of Buterin's "A Proof of Stake Design Philosophy".* [Online] Available at: https://tuurdemeester.medium.com/critique-of-buterins-a-proof-of-stake-design-philosophy-49fc9ebb36c6 [Zugriff am Juni 2022].

Eberhardt, J. & Tai, S., 2017. *On or Off the Blockchain? Insights on Off-Chaning Computation and Data,* TU Berlin: s.n.

Elmasri, R. & Navathe, B. S., 2002. *Grundlagen von Datenbanksystemen.* 3. Auflage Hrsg. Deutschland: Pearson Deutschland.

Faber, E. v. & Behnsen, W., 2018. *Wichtige Begriffe der IT-Sicherheit,* s.l.: s.n.

Fill, H.-G. & Meier, A., 2020. *Blockchain kompakt.* s.l.:Springer.

Frauenhofer-Gesellschaft, 2017. *Blockchain und Smart Contracts - Technologien, Forschungsfragen und Anwendung.* [Online] Available at: https://www.aisec.fraunhofer.de/content/dam/aisec/Dokumente/Publikationen/Studien_Tech

Reports/deutsch/Fraunhofer-Positionspapier_Blockchain-und-Smart-Contracts.pdf
[Zugriff am Februar 2022].

Geiling, L., 2016. *Distributed Ledger: Die Technologie hinter den virtuellen Währungen am Beispiel der Blockchain.* [Online]
Available at: https://www.bafin.de/dok/7845804
[Zugriff am März 2022].

Gentemann, L. & Heidkamp, P., 2021. *Cloud-Monitor 2021,* s.l.: bitkom.

Gerken, W., 2016. *Datenbanksysteme für Dummies.* Deutschland: WILEY-VCH GmbH & Co.KGaA.

Hansen, M., 2012. *Vertraulichkeit und Integrität von Daten und IT-Systemen im Cloud-Zeitalter.* [Online]
Available at: https://doi.org/10.1007/s11623-012-0149-8
[Zugriff am Februar 2022].

Hellwig, D., Karlic, G. & Huchzermeier, A., 2021. *Entwickeln Sie ihre eigene Blockchain.* Berlin: Springer Gabler.

Hertz, T., 2003. *Klassifizierung und Bewertung von Persistenz-Management Technologien in J2EE Architekturen unter besonderer Betrachtung von Skalierbarkeit und Ausfallsicherheit,* München: Technische Universität München, Fakultät für Informatik.

Hyperledger Besu, C., 2022. *System requirements for private networks.* [Online]
Available at: https://besu.hyperledger.org/en/stable/HowTo/Get-Started/System-Requirements/System-Requirements-Private/
[Zugriff am Juni 2022].

Hyperledger Fabric, D., 2017. *Hyperledger Fabric Docs.* [Online]
Available at: https://hyperledger-fabric.readthedocs.io/en/
[Zugriff am März 2022].

Isler, M., 2022. *Blockchain Infrastructure Requirments (Software & Hardware)*. [Online] Available at: https://imiblockchain.com/blockchain-infrastructure-requirements/ [Zugriff am Juni 2022].

Karaarslan, E. & Konacaklı, E., 2020. *Data Storage in the Decentralized World: Blockchain and Derivatives.* [Online] Available at: https://arxiv.org/abs/2012.10253 [Zugriff am Februar 2022].

kinsta, 2022. *AWS Marktanteil.* [Online] Available at: https://kinsta.com/de/aws-marktanteil/ [Zugriff am Juni 2022].

L. Rabe, 2021. *Statista, Anzahl der Internetnutzer weltweit bis 2021*. [Online] Available at: https://de.statista.com/statistik/daten/studie/805920/umfrage/anzahl-der-internetnutzer-weltweit/#:~:text=Die%20Zahl%20der%20Internetnutzer%20weltweit,rund%202%2C73%20Milliarden%20gestiegen.
[Zugriff am Mai 2022].

Learn, B., 2022. *Peer-to-Peer Blockchain Networks: The Rise of P2P Crypto Exchanges.* [Online]
Available at: https://learn.bybit.com/bybit-p2p-guide/peer-to-peer-blockchain-network/ [Zugriff am März 2022].

Lewin, M., Dogan, A., Schwarz, J. & Fay, A., 2019. *Distributed-Ledger-Technologien und Industrie 4.0.* Berlin, Springer-Verlag.

Lips, P. et al., 2015. *Technischer Fortschritt im Gesundheitswesen: Quelle für Kostensteigerungen* , Deutschland: Ausschusses für Bildung, Forschung und Technikfolgeabschätzung, Deutscher Bundestag.

Luu, L. et al., 2016. *Making Smart Contracts Smarter,* Singapore: s.n.

Mahlmann, P. & Schindelhauer, C., 2007. *Peer-to-Peer-Netzwerke.* Berlin: Springer.

Meier, A. & Fill, H.-G., 2020. *Blockchain Grundlagen, Anwendungsszenarien und Nutzungspotenziale.* Fribourg: Springer Vieweg.

Merkel, M., 2009. Identity Management und Single-Sign-On. In: *Netzsicherheit und Hackerabwehr.* Universität Karlsruhe: Institut of Telematics, p. 138.

Merkle, R., 1987. A Digital Signature Based on a Conventional Encryption Function. In: *Proceedings Advances in Cryptology – CRYPTO.* Carlifornia: s.n., p. 369–378.

Michael, M., 2011. *Discovering P2P,* San Francisco: s.n.

Mock, D. M., Sylla, K.-H. & Hecker, D. D., 2014. Skalierbarkeit und Architektur von Big-Data-Anwendungen. *OBJEKTSpektrum Ausgabe IT-Trends.*

Nakamoto, S., 2008. *Bitcoin.org.* [Online] Available at: https://bitcoin.org/bitcoin.pdf [Zugriff am 23 02 2022].

Pizette, L. & Cabot, T., 2012. *Database as a Service: A Marketplace Assessment.* [Online] Available at: https://www.mitre.org/sites/default/files/pdf/cloud_database_service_dbaas.pdf [Zugriff am April 2022].

Pohlmann, N., 2019. *Cyber-Sicherheit.* Wiesbaden: Springer Vieweg.

Qiao Yan, F. R. Y. S. M. I. Q. G. a. J. L., 2016. *Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges.* s.l., s.n., pp. 602-622.

Quorum, J. M., 2022. *Tessera.* [Online] Available at: https://docs.tessera.consensys.net/en/stable/ [Zugriff am Juni 2022].

Ray, C., 2009. *Distributed database systems.* India: Pearson Education India.

Rehfeld, D., 2020. *Handbuch Digitalisierung in Staat und Verwaltung pp 1-11,* Aachen: s.n.

Rost, M. & Pfitzmann, A., 2009. *Datenschutz-Schutzziele,* s.l.: s.n.

Rügheimer, H., 2019. Zwei Faktoren für mehr Sicherheit. *Schmerzmedizin 35.*, 22 Juli, pp. 53-53.

S.-D., L., Q., L., M., C. & Z, a. Z., 2004. *A Practical Distributed Mutual Exclusion.* Berlin, Springer-Verlag, pp. 11-12.

Sanders, J., 2019. *TechRepublic - Backblaze opens data center in Amsterdam, retains same pricing as US.* [Online]
Available at: https://www.techrepublic.com/article/backblaze-opens-data-center-in-amsterdam-retains-same-pricing-as-us/
[Zugriff am Juni 2022].

Schonschek, O. & Witmer-Goßner, E., 2020. *Datenübermittlung in die USA - Die Rolle des Cloud-Standorts für den Datenschutz.* [Online]
Available at: https://www.cloudcomputing-insider.de/die-rolle-des-cloud-standorts-fuer-den-datenschutz-a-963676/
[Zugriff am April 2022].

Schurtenberger, M., 2022. *Öffentliche vs. private Blockchains: Warum öffentliche Blockchains die Zukunft sind.* [Online]
Available at: https://www.bitcoinsuisse.com/de/outlook/why-public-blockchains-are-the-future-2
[Zugriff am 23 02 2022].

Schütte, J. et al., 2017. *Blockchain und Smart Contracts - Technologien, Forschungsfragen und Anwendungen*, Bayerruth: s.n.

Shehri, W. A., 2013. *Cloud Database - Database as a Service.* Australien, Academia, pp. 1-13.

Songze Li, M. Y. C.-S. Y. A. S. A. S. K. a. P. V., 2020. *PolyShard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously.* s.l., s.n., pp. PP. 1-1..

Tanenbaum, A. S. & Steen, M. v., 2018. *Distributed Systems.* s.l.:Pearson Education.

Tanenbaum, A. S. & Wetherhall, D. J., 2011 5th ed. . *Computer Networks.* Boston: Pearson Education .

Thießen, F., 2020. *Öffentliche vs. Private Blockchains in der Finanzwirtschaft,* Chemnitz: s.n.

Toyka-Seid, C. & Schneider, G., 2022. *Bundeszentrale für politische Bildung.* [Online] Available at: https://www.bpb.de/kurz-knapp/lexika/das-junge-politik-lexikon/320107/dezentralisierung/
[Zugriff am März 2022].

TRUST 2014, 2014. *Trust and Trustworthy Computing.* Kreta, Griechenland, Springer International Publishing, p. 115.

Vossen, G., 2008. *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme.* 5. Auflage Hrsg. München: Oldenbourg Verlag München Wien.

Xiao, Z. et al., 2018. *EMRShare: A Cross-organizational Medical Data Sharing and Management Framework Using Permissioned Blockchain.* s.l., IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS).

Ye, C. et al., 2018. *Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting.* Shanghai, DSA, pp. 15-24.

Zargar, S. T., Joshi, J. & Tipper, D., 2013. *A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks.* s.l., s.n., p. pp. 2046–2069.

Zehnder, C. A., 2013. *Informationssysteme und Datenbanken.* s.l.:Springer-Verlag.

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.
Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Hecht

Vorname: Rudolf

dass ich die vorliegende Bachelorarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Vergleich einer Private Permissioned Blockchain mit traditionellen Datenbanksystemen

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der -bitte auswählen- ist erfolgt durch:

Hamburg	04.07.2022	[redacted]
Ort	Datum	Unterschrift im Original