



Eine Bachelorthesis zur Erlangung des  
akademischen Grades Bachelor of Science (B.Sc.)  
zum Thema:

**Anforderungsanalyse und Design einer  
Arbitrage-Trading Zielarchitektur für  
Kryptowährungen**

**Ekrem Bugday**  
**Matrikelnummer: 1325425**

Hauptreferent: Prof. Dr. rer. nat. Karsten Weronek

Korreferent: Prof. Dr. Martin Simon

Fachbereich 2: Informatik und Ingenieurwissenschaften

Frankfurt University of Applied Sciences

Frankfurt am Main, Deutschland

04. April 2023

# Eidesstattliche Erklärung

Hiermit versichere ich, Ekrem Bugday, an Eides statt, dass ich die vorliegende Arbeit mit dem Titel „Anforderungsanalyse und Design einer Arbitrage-Trading Zielarchitektur für Kryptowährungen“, selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter der Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form an irgendeiner Stelle als Prüfungsleistung vorgelegt worden. Die Anmeldung beim Prüfungsamt der Frankfurt University of Applied Sciences für die Prüfung ist erfolgt.

A handwritten signature in black ink, appearing to read 'Bugday', with a long horizontal stroke extending to the right.

Ekrem Bugday

Frankfurt am Main, 04.04.2023

# Danksagung

Zuallererst möchte ich mich ganz herzlich bei meinem Hauptreferenten, Prof. Dr. rer. nat. Karsten Weronek, für seine kontinuierliche Unterstützung, hervorragende Fachkompetenz und wertvolle Anregungen während der gesamten Bachelorthesis bedanken.

Des Weiteren gilt mein Dank meinem Korreferenten, Prof. Dr. Martin Simon, für seine hilfreichen Ratschläge und das aufschlussreiche Feedback, welches einen wesentlichen Beitrag zur Qualität meiner Arbeit geleistet hat.

Ein besonderer Dank gebührt meiner Familie, die mir in jeder Hinsicht den Rücken gestärkt und mich ermutigt hat, immer nach Höherem zu streben. Ihre Liebe, Unterstützung und Geduld waren unersetzlich.

Schließlich möchte ich mich bei allen Korrekturlesern bedanken, die sich die Zeit genommen haben, meine Arbeit aufmerksam zu lesen und konstruktive Kritik sowie wertvolle Anregungen zu geben. Eure Unterstützung war essenziell für die Fertigstellung meiner Bachelorthesis.

In Dankbarkeit,  
Ekrem Bugday

# Abstract

In dieser Bachelorthesis wird die Anforderungsanalyse und das Design einer Arbitrage-Trading Zielarchitektur für Kryptowährungen mit Fokus auf das Handelspaar BTC/USDT untersucht. Der Schwerpunkt liegt auf der Identifizierung und Systematisierung von Anforderungen, die für den erfolgreichen Einsatz von automatisierten Arbitrage-Strategien im Kryptowährungshandel notwendig sind. Die Arbeit betrachtet verschiedene Aspekte, wie die Funktionsweise von Kryptowährungsbörsen, Preisbildung, technische Herausforderungen und Risiken, die mit der Krypto-Arbitrage verbunden sind.

Die Zielarchitektur wird unter Verwendung von Python als Programmiersprache entwickelt, um von der geringen Komplexität, Flexibilität und Skalierbarkeit der Programmiersprache zu profitieren. Darüber hinaus werden relevante Bibliotheken und Werkzeuge zur Implementierung sowie Optimierung von Arbitrage-Strategien untersucht. Die Arbeit befasst sich zudem mit der Integration und Anpassung der Zielarchitektur an verschiedene Kryptowährungsbörsen, um die Effizienz und Profitabilität der Arbitrage-Trading-Anwendung für das Handelspaar BTC/USDT zu maximieren.

Schließlich wird die entwickelte Arbitrage-Trading-Anwendung hinsichtlich ihrer Profitabilität und Leistungsfähigkeit im Kontext des BTC/USDT-Handelspaars analysiert. Die Ergebnisse dieser Analyse dienen dazu, die Praktikabilität der vorgeschlagenen Zielarchitektur zu bewerten und mögliche Verbesserungen und Erweiterungen für zukünftige Forschungsarbeiten zu identifizieren.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation	1
1.2	Zielsetzung	1
1.3	Gliederung	2
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Kryptowährungen	4
2.1.1	Definition	4
2.1.2	Funktionsweise	4
2.1.3	Vor- und Nachteile gegenüber traditionellen Finanzsystemen	5
2.1.4	Bitcoin (BTC)	6
2.1.5	Tether (USDT)	6
2.2	Kryptowährungsbörsen	7
2.2.1	Funktionsweise auf Zentralisierte Börsen (CEX)	8
2.3	Arbitrage im Kryptohandel	9
2.3.1	Verschiedene Arten von Arbitrage	10
2.3.2	Herausforderungen und Risiken	10
2.4	Historische Tick-Daten	11
2.4.1	Relationale Datenbanken – PostgreSQL	12
2.5	Verwandte Arbeiten	13
<b>3</b>	<b>Anforderungsanalyse</b>	<b>14</b>
3.1	Anforderungsanalyse	14
3.1.1	Funktionale Anforderungen	14
3.1.2	Nicht-funktionale Anforderungen	16
<b>4</b>	<b>Vorgehensweise / Ansatz</b>	<b>18</b>
4.1	Datensammlung	18
4.2	Datenverarbeitung	18
4.3	Datenanalyse	20

<b>5 Design</b>	<b>22</b>
5.1 Einleitung . . . . .	22
5.2 Technische Details . . . . .	22
5.3 Systemarchitektur . . . . .	23
5.4 Programmablauf . . . . .	25
5.4.1 Systemkomponenten . . . . .	26
<b>6 Analyse und Test</b>	<b>28</b>
6.1 Funktionale Tests . . . . .	28
6.2 Nicht-funktionale Tests . . . . .	35
6.3 Testumgebung . . . . .	40
<b>7 Bewertung</b>	<b>41</b>
7.1 Bewertung und Fazit . . . . .	41
7.1.1 Bewertung . . . . .	41
7.1.2 Fazit . . . . .	42
7.2 Ausblick . . . . .	42
7.2.1 Verbesserung der Handelslogik . . . . .	42
7.2.2 Einsatz von Websockets . . . . .	42
7.2.3 Integration weiterer Börsen . . . . .	42
7.3 Zusammenfassung . . . . .	43
<b>8 Anhang</b>	<b>45</b>

# Abbildungsverzeichnis

5.1 UML-Klassendiagramm der Architektur . . . . .	24
6.1 Orderbuchabfrage für Bitstamp, Coinbase und Binance . . . . .	29
6.2 Niedrigste Briefkurs (Ask)- und höchste Geldkurs (Bid)kurse . . . . .	30
6.3 Ask und Bidkurse nach Gebühr und Menge . . . . .	30
6.4 Identifizierung von Arbitragemöglichkeiten . . . . .	32
6.5 .log-Datei ohne Arbitragemöglichkeiten . . . . .	33
6.6 .log-Datei, die Arbitragemöglichkeiten protokolliert . . . . .	34
6.7 Ausgabe nach erfolgreichem Klonen des GitHub-Repositorys . . . . .	35
6.8 Ausgabe nach erfolgreichem Update . . . . .	36
6.9 Quelltextänderung, um Coinbase-Teilausfall zu simulieren . . . . .	37
6.10 Ausgabe nach Simulation des Coinbase-Teilausfalls . . . . .	37
6.11 Erfolgreiche Fehlerbehandlung eines Börsenausfalls . . . . .	38
6.12 Ausgabe der Lauffähigkeit auf Windows . . . . .	38
6.13 Ausgabe der Lauffähigkeit auf macOS . . . . .	39

# Abkürzungsverzeichnis

**API** Application Programming Interface.

**Ask** Briefkurs.

**Bid** Geldkurs.

**BTC** Bitcoin.

**BTC/USDT** Handelspaar Bitcoin/Tether.

**CEX** Zentralisierte Börsen.

**DeFi** Dezentralisierte Finanzsysteme.

**DEX** Dezentralisierte Börsen.

**P2P** Peer-to-Peer.

**PoC** Proof of Concept.

**PoS** Proof-of-Stake.

**PoW** Proof-of-Work.

**REST** Representational State Transfer.

**RoT** Return of Trade.

**SQL** Structured Query Language.

**TWh** Terawattstunde.

**UI** Benutzeroberfläche.

**USD** US-Dollar.

**USDT** Tether.



# Kapitel 1

## Einleitung

### 1.1 Motivation

In den letzten Jahren hat der Kryptowährungsmarkt eine enorme Popularität gewonnen und zu renditestarken Investitionen in diese Vermögenswerte geführt. Die hohe Volatilität dieser digitalen Vermögenswerte hat ein zunehmendes Interesse von Anlegern und Händlern geweckt, die nach innovativen Handelsstrategien suchen, um Gewinne zu erzielen. Eine dieser Strategien ist das Arbitrage-Trading, dessen Ziel es ist, von Preisunterschieden auf verschiedenen Börsen zu profitieren. Die Strategie besteht darin, ein Finanzinstrument an einer Börse zu kaufen und es an einer Alternativbörse zu einem höheren Preis zu verkaufen [1]. Da der Kryptomarkt ein noch junger, schnell wachsender sowie unregulierter Markt ist [2] und Preisunterschiede häufiger als in traditionellen Finanzmärkten zu beobachten sind, stellt dieser eine vielversprechende Möglichkeit für das Arbitrage-Trading dar. Die Motivation für diese Arbeit liegt darin, eine effektive und skalierbare Lösung für das Arbitrage-Trading von Kryptowährungen zu entwickeln und so einen Beitrag zur Verbesserung der Handelseffizienz in diesen Märkten zu leisten.

### 1.2 Zielsetzung

Die Zielsetzung dieser Bachelorthesis ist es, eine Arbitrage-Trading Zielarchitektur für Kryptowährungen zu entwerfen und dabei sowohl funktionale als auch nicht-funktionale Anforderungen zu berücksichtigen. Um dieses Ziel zu erreichen, werden die folgenden Schritte unternommen:

1. Identifikation von Anforderungen: Eine gründliche Anforderungsanalyse wird durchgeführt, um die notwendigen Parameter und Datenquellen für eine erfolgreiche Arbitrage-Strategie zu ermitteln und somit eine solide Grundlage für das Design der Zielarchitektur zu schaffen.

2. Design der Zielarchitektur: Basierend auf den ermittelten Anforderungen wird eine passende Architektur entworfen, die eine effiziente und skalierbare Umsetzung von Arbitrage-Trading-Strategien ermöglicht.
3. Auswahl von Technologien und Frameworks: Im Rahmen des Designprozesses werden geeignete Technologien und Frameworks identifiziert, die sowohl die Umsetzung der Zielarchitektur als auch die Einhaltung von nicht-funktionalen Anforderungen, wie beispielsweise Leistung, Sicherheit und Skalierbarkeit, unterstützen.
4. Analyse von Risiken und Herausforderungen: Es wird eine Untersuchung möglicher Risiken und Herausforderungen bei der Implementierung der entwickelten Zielarchitektur durchgeführt, um sicherzustellen, dass die entwickelte Lösung praktikabel bzw. tragfähig ist.
5. Evaluation der Profitabilität: Schließlich wird die entwickelte Arbitrage-Trading Zielarchitektur hinsichtlich ihrer Profitabilität evaluiert. Dabei werden verschiedene Szenarien und Parameter analysiert, um die Rentabilität der Anwendung in unterschiedlichen Marktbedingungen zu untersuchen.

Die erfolgreiche Umsetzung dieser Ziele wird dazu beitragen, das Verständnis von Arbitrage-Trading-Strategien in Kryptowährungsmärkten zu vertiefen und neue Möglichkeiten für Anleger bzw. Händler zu erschließen. Zudem wird durch die Evaluation der Profitabilität sichergestellt, dass die entwickelte Zielarchitektur nicht nur technisch umsetzbar ist, sondern auch einen finanziellen Mehrwert für Anwender bietet. Dadurch können die gewonnenen Erkenntnisse und die entwickelte Lösung einen positiven Einfluss auf die Handelspraktiken in der Kryptowährungsbranche haben und zur weiteren Reifung und Stabilisierung dieser aufstrebenden Märkte beitragen.

## 1.3 Gliederung

Nach dieser Einleitung werden im Kapitel [2](#) die für das Verständnis der Thematik dieser Thesis nötigen Grundlagen erläutert. Im Anschluss daran wird in Kapitel ?? zuerst beschrieben, weshalb sich in dieser Arbeit auf bestimmte Architekturen und Technologien fokussiert wurde und um welche es sich dabei handelt. Außerdem wird in Kapitel ?? im Detail auf die Best Practices zur Implementierung von OAuth 2.0 und OIDC eingegangen und anschließend das verwendete Analyseverfahren erklärt.

In Kapitel ?? werden die Architekturen sowie der Entwicklungsprozess der Prototypenwendungen im Detail erklärt. Im Anschluss daran werden im Kapitel ?? die Best Practices und WebAuthn auf deren Effektivität durch das vorher beschriebene Analyseverfahren analysiert. Abschließend liefert Kapitel [7](#) ein Fazit über die Effektivität der Best Practices und den Sicherheitsvorteil von WebAuthn inklusive eines Ausblicks.

Nach dieser Einleitung werden im Kapitel [2](#) die für das Verständnis der Thematik dieser Thesis nötigen Grundlagen erläutert. Im Anschluss werden in Kapitel [3](#) die funktionalen und nicht-funktionalen Anforderungen an die Architektur gestellt. In Kapitel [4](#) das Vorgehen der Datenaufbereitung und die Analyse der historischen Tick-Daten beschrieben.

# Kapitel 2

## Grundlagen

### 2.1 Kryptowährungen

#### 2.1.1 Definition

Kryptowährungen stellen digitale Vermögenswerte dar, die auf kryptografischen Funktionen basieren und als Tauschmittel fungieren. Im Gegensatz zu herkömmlichen Fiat-Währungen wie dem Euro oder Dollar sind sie nicht an eine zentrale Instanz wie eine Zentralbank oder Regierung gebunden [3],[4]. Die Ursprünge von Kryptowährungen lassen sich auf das revolutionäre Werk von Nakamoto [5] zurückführen, welcher erstmals das Bitcoin-Protokoll und dessen dezentralisierte Infrastruktur beschrieb. Seither sind eine Vielzahl alternativer Kryptowährungen, sogenannter Altcoins, entstanden, welche verschiedene technologische Ansätze und Anwendungsbereiche aufweisen [6],[7].

#### 2.1.2 Funktionsweise

Die grundlegende Technologie hinter Kryptowährungen ist die Blockchain, welche ein dezentrales Hauptbuch nutzt, um Transaktionen zwischen zwei Teilnehmern in einer sicheren, unveränderlichen und persistenten Reihenfolge zu verarbeiten [8]. Die Blockchain besteht aus einer Abfolge von Blöcken, die Transaktionen speichern und durch kryptografische Hash-Funktionen miteinander verknüpft sind [9],[10],[11]. Jeder Block enthält eine Referenz auf den vorherigen Block und die kryptografischen Hash-Werte des vorherigen Blocks, um eine unveränderliche Abfolge von Transaktionen sicherzustellen [5],[12].

Durch die dezentrale Struktur der Blockchain besitzen alle Netzwerkteilnehmer eine Kopie des Hauptbuchs. Die Validierung und Überprüfung der Transaktionen auf der Blockchain erfolgt durch die Netzwerkteilnehmer selbst, ohne dass eine zentrale Instanz benötigt wird. Die Anwendung von Konsensmechanismen wie **Proof-of-Work (PoW)** und **Proof-of-Stake (PoS)** dient dazu, die Netzwerkteilnehmer dazu zu bewegen, ihre

Ressourcen und Rechenleistung dem Netzwerk zur Verfügung zu stellen und damit zur Aufrechterhaltung der Netzwerkintegrität beizutragen. Dabei werden sie durch monetäre Anreize in Form von Entlohnungen motiviert [13]. Positive Nebeneffekte für das Netzwerk ist das verringerte Risikomaß möglicher Datenmanipulationen sowie die Schaffung erhöhter Transparenz und Sicherheit [14],[15].

### 2.1.3 Vor- und Nachteile gegenüber traditionellen Finanzsystemen

Kryptowährungen weisen im Vergleich zu traditionellen Finanzsystemen eine Vielzahl von Vorteilen auf. Eine dieser Eigenschaften ist die Möglichkeit, **Peer-to-Peer (P2P)**-Transaktionen zwischen Benutzern durchzuführen, ohne dass eine zentrale Autorität erforderlich ist. Weitere Vorteile umfassen niedrigere Transaktionskosten, schnellere grenzüberschreitende Transaktionen, eine gesteigerte finanzielle Inklusion sowie Zensurresistenz [16]. Ein weiterer wesentlicher Vorteil besteht darin, dass Kryptowährungen immun gegen die Infiltration von Falschgeld sind [17]. Darüber hinaus ermöglichen sie die Entwicklung neuer Finanzinstrumente und -dienstleistungen, wie beispielsweise **Dezentralisierte Finanzsysteme (DeFi)** und Smart Contracts, die auf der Ethereum-Blockchain eingeführt wurden [18]. Die genannten Vorteile haben das Potenzial, den Finanzsektor grundlegend zu verändern und eine disruptive Technologie zu schaffen [19].

Trotz der Vorteile bergen Kryptowährungen zugleich Herausforderungen und Risiken. Ein wesentlicher Nachteil von Kryptowährungen stellt die Umweltbelastung dar. Insbesondere **PoW**-basierte Kryptowährungen verbrauchen signifikante Mengen an Energie. Der jährliche Energieverbrauch war im Jahr 2018 mit 61,3 **Terawattstunde (TWh)** vergleichbar mit dem der Schweiz und weist eine steigende Tendenz auf [20]. Für das Jahr 2022 wird der Verbrauch von 150 **TWh** geschätzt, mit einem erwarteten Anstieg um weitere sechs **TWh** bis Mitte 2023 [21]. Ein weiterer Nachteil besteht darin, dass der Handel mit Kryptowährungen zum größten Teil unreguliert ist und „Kryptowährungen als der Wilde Westen gelten.“ [22]. Die Auswirkungen des unregulierten Kryptowährungsmarktes sind zum einen, dass dieser starken Preisschwankungen ausgesetzt und somit äußerst volatiler Natur ist. Im Vergleich zu traditionellen Währungen, die in der Regel von Zentralbanken reguliert werden, um Preisstabilität zu gewährleisten, erschweren diese Schwankungen die Verwendung von Kryptowährungen als stabiles Tausch- und Wertaufbewahrungsmittel [23]. Zum anderen werden hierdurch Anreize für dolose Handlungen geschaffen [24]. So lag u. a. das weltweite Geldwäsche-Volumen in diesem Markt bei 8,6 Milliarden US-Dollar [25].

Im Vergleich zu traditionellen Währungen, die in der Regel von Zentralbanken reguliert werden, um Preisstabilität zu gewährleisten, erschweren diese Schwankungen die Verwendung von Kryptowährungen als stabiles Tausch- und Wertaufbewahrungsmittel. Zum anderen werden hierdurch Anreize für dolose Handlungen geschaffen. So lag das weltweite Geldwäsche-Volumen in diesem Markt bei 8,6 Milliarden

#### 2.1.4 **BTC**

Der Bitcoin wurde 2008 von einer Person oder einer Gruppe unter dem Pseudonym Satoshi Nakamoto in einem Whitepaper vorgestellt [5] und im Januar 2009 wurde die erste Bitcoin-Transaktion durchgeführt [26]. Heutzutage hat er als Leitwährung in der Kryptowelt [27] die höchste Marktkapitalisierung [27]. Beim Bitcoin handelt sich um eine dezentrale Kryptowährung, die keine physische Form hat und vollständig elektronisch übertragen wird. Die Währung basiert auf der Blockchain-Technologie und ist auf P2P-Transaktionen angewiesen [13].

Bitcoins werden durch einen Prozess namens Mining generiert. Dabei werden komplexe mathematische Probleme gelöst, um neue Blöcke zur Blockchain hinzuzufügen und Transaktionen zu bestätigen. Als Belohnung für diesen Prozess erhalten Miner eine festgelegte Anzahl von Bitcoins. Die Menge an Bitcoins, die als Belohnung ausgegeben wird, halbiert sich aufgrund des Algorithmus alle 210.000 Blöcke, was etwa alle vier Jahre passiert [5], [28].

Die Anzahl der Bitcoins ist insgesamt auf circa 21 Millionen begrenzt, was zur Folge hat, dass nach Erreichen dieser Menge keine weiteren Bitcoins mehr generiert werden können. Diese Eigenschaften sind im Quellcode [29] der Bitcoin-Blockchain festgehalten. Änderungen des Quellcodes können nur durch Konsens der Teilnehmer im Netzwerk geschaffen werden [30].

#### 2.1.5 **Tether (USDT)**

Tether repräsentiert eine Kryptowährungsklasse, die als Stablecoin bezeichnet wird und darauf abzielt, den Wert des US-Dollars (USD) digital abzubilden [31]. Seit seiner Einführung im Jahr 2014 hat Tether erhebliche Bedeutung im Kryptowährungssystem erlangt, indem es eine stabile digitale Alternative zum US-Dollar (USD) bietet, wobei der Wechselkurs zwischen Tether (USDT) und USD annähernd 1:1 bleibt, denn laut Tether Limited, dem Unternehmen hinter USDT, ist jeder emittierte USDT-Token durch entsprechende Reserven gedeckt [32]. Die Entwicklung von Stablecoins wie USDT hat eine wichtige Debatte über ihre Rolle im Kryptowährungsmarkt und ihre potenziellen Auswirkungen auf das traditionelle Finanzsystem ausgelöst [33].

Der wesentliche Unterschied zwischen **USDT** und **USD** ist, dass **USDT** ein digitaler Vermögenswert ist, der auf einer Blockchain gespeichert und übertragen wird, während es sich bei **USD** um eine konventionelle Fiat-Währung handelt, die sowohl in physischer, also in Banknoten und Münzen, als auch in elektronischer Form, also als Buchgeld auf Bankkonten, existiert. **USDT** bietet im Vergleich zum **USD** mehrere Vorteile, wie beispielsweise schnellere Transaktionsgeschwindigkeiten, geringere Transaktionsgebühren und die Möglichkeit, rund um die Uhr und ortsunabhängig gehandelt zu werden [34], [35].

Ein Grund, warum Kryptowährungsbörsen **USDT** statt **USD** für den Handel nutzen, liegt in der regulatorischen Komplexität, die mit Fiat-Währungen verbunden ist. Die Integration von **USD** in Kryptowährungsbörsen erfordert die Einhaltung strenger regulatorischer Anforderungen, einschließlich des Erwerbs einer Banklizenz oder der Zusammenarbeit mit einer lizenzierten Bank. Durch die Verwendung von **USDT** können Börsen diesen regulatorischen Hürden ausweichen und dennoch ein stabiles digitales Äquivalent des **USD** für ihre Nutzer bereitstellen [36].

## 2.2 Kryptowährungsbörsen

Kryptowährungsbörsen sind essenziell für den Handel und die Preisbildung von Kryptowährungen und dienen als digitale Marktplätze, auf denen Käufer und Verkäufer Kryptowährungen gegen Fiat-Währungen oder andere Kryptowährungen tauschen können [37]. Verschiedene Aspekte wie Handelsgebühren, Sicherheit, Handelsvolumen, verfügbare Handelspaare und Benutzerfreundlichkeit unterscheiden sich zwischen den Börsen [38], [13].

Im Wesentlichen gibt es zwei Haupttypen von Kryptowährungsbörsen: **CEX** und **Dezentralisierte Börsen (DEX)**. Beispiele für dezentralisierte Börsen sind unter anderem Uniswap, SushiSwap und PancakeSwap, wohingegen die Börsen Binance, Bitstamp und Coinbase **CEX** repräsentieren. Während **CEX** von einer zentralen Institution betrieben werden, basieren **DEX** auf Blockchain-Technologie und ermöglichen den **P2P**-Handel. Grundsätzlich agieren **CEX** wie traditionelle Börsen, z. B. die Frankfurter Börse, indem sie den Handel zwischen Käufern und Verkäufern organisieren und erleichtern. **CEX** stellen mit ca. 95 % des gehandelten Handelsvolumens die am häufigsten genutzten Handelsplattformen dar. Diese Börsen bieten in der Regel einen einfachen Zugang, eine breite Palette von Handelspaaren sowie Liquidität [39], können aber anfällig für Hacks und regulatorische Probleme sein [13] [40]. Des Weiteren sind **CEX** in der Lage, Kundeneinlagen in riskante Spekulations- und Kreditgeschäfte zu investieren. Ein derartiges Vorgehen führte beispielsweise im Fall der Börse FTX zur verheerenden

Insolvenz in Milliardenhöhe, die vielen Kunden bedeutende Verluste bescherte. [41], [42].

[DEX] basieren auf der Blockchain-Technologie und ermöglichen den [P2P]-Handel. Sie nutzen die Vorteile der Blockchain-Technologie und bieten erhöhte Sicherheit und Anonymität, da Benutzer die volle Kontrolle über ihre Vermögenswerte behalten und keine persönlichen Informationen an eine zentrale Institution übermitteln müssen. Darüber hinaus sind [DEX] tendenziell resistenter gegen Zensur und regulatorische Eingriffe. Jedoch weisen dezentralisierte Börsen ebenso Nachteile auf, wie z. B. deutlich geringere Liquidität und eine in der Regel weniger benutzerfreundliche Handelsoberfläche im Vergleich zu zentralisierten Börsen [39]. Obwohl DEX tendenziell sicherer sind als [CEX] [13], sind sie dennoch nicht vollständig immun gegen Angriffe oder Hacks, insbesondere aufgrund von Schwachstellen in den zugrunde liegenden Smart Contracts [43], siehe Liste von aktuellen Hacks und Exploits [44]. Im Kontext des Arbitragehandels sind die größten Nachteile von [DEX], dass zunächst der Handel mit Fiat-Währungen nicht möglich ist und dass der Handel auf diesen Plattformen in der Regel deutlich langsamer ist als auf [CEX].

### 2.2.1 Funktionsweise auf [CEX]

Der Handel auf den bekanntesten [CEX] basiert marktüblich auf dem Prinzip von Angebot und Nachfrage, wobei Käufer und Verkäufer ihre gewünschten Preise und Mengen für den Handel von Kryptowährungen festlegen [45], [38]. Im Folgenden werden die Funktionsweisen von Kryptowährungsbörsen und deren wichtigsten Elemente detailliert erläutert:

1. Orderbuch: Das Orderbuch ist ein zentrales Element einer Kryptowährungsbörse. Es ist eine Liste aller offenen Kauf- und Verkaufsaufträge, die von den Benutzern der Börse platziert wurden. Das Orderbuch ist in der Regel in zwei Teile unterteilt: das Bid-Orderbuch (Kaufaufträge) und das Ask-Orderbuch (Verkaufsaufträge). Jeder Eintrag im Orderbuch enthält den Preis, zu dem ein Benutzer bereit ist, eine bestimmte Menge einer Kryptowährung zu kaufen oder zu verkaufen. Die Aufträge werden im Orderbuch nach ihrem Preis sortiert, wobei die höchsten Bid-Aufträge und die niedrigsten Ask-Aufträge an der Spitze der Liste platziert sind [45]. Die Differenz zwischen dem höchsten Bid-Auftrag und dem niedrigsten Ask-Auftrag wird als Spread bezeichnet.
2. Handelspaare: Kryptowährungsbörsen bieten in der Regel verschiedene Handelspaare an, die es Benutzern ermöglichen, eine Kryptowährung gegen eine andere zu tauschen. Ein Handelspaar besteht aus zwei Währungen, beispielsweise [BTC] und [USDT] im Handelspaar [Handelspaar Bitcoin/Tether (BTC/USDT)]. In die-



sem Fall wird Bitcoin als Basiswährung und Tether als Zielwährung bezeichnet [46].

3. Auftragsarten: Es existieren verschiedene Auftragsarten, die Benutzer platzieren können. Es wird unterschieden zwischen Marktaufträgen, die für eine sofortige Ausführung des Kauf- oder Verkaufsauftrags zum aktuellen Marktpreis stehen, Limitaufträge, die ausgeführt werden, sobald ein vom Benutzer festgelegter Preis erreicht ist. Zusätzlich gibt es Stop-Loss Aufträge, welche ausgeführt werden, sobald ein bestimmter Schwellenwert erreicht wird. Letztere werden zur Begrenzung von Verlusten eingesetzt und werden beim Arbitrage-Trading nicht genutzt [46].
4. Maker und Taker: Die Begriffe Maker und Taker beziehen sich auf die Rolle, die ein Benutzer beim Platzieren eines Kauf- oder Verkaufsauftrags einnimmt. Ein Maker ist ein Benutzer, der einen Limitauftrag im Orderbuch platziert, wodurch Liquidität hinzugefügt wird. Ein Taker hingegen ist ein Benutzer, der entweder Markt- oder Limitaufträge platziert, die sofort ausgeführt werden, indem sie einen bestehenden Auftrag im Orderbuch ausführt und somit Liquidität entnimmt [47].
5. Handelsgebühren: Börsen haben in der Regel Handelsgebühren, die für die Ausführung von Trades erhoben werden. Dabei sind die Gebühren normalerweise als Prozentsatz des Handelsvolumens festgelegt. Oft unterscheiden sich die Gebühren für Maker und Taker, wobei Maker in der Regel geringere Gebühren zahlen, da Börsen ein möglichst großes Orderbuch anstreben [48]. Beim Arbitrage-Trading agiert der Anleger hauptsächlich als Taker, da man die von Makern bereitgestellten Aufträge instant erfüllen möchte.

## 2.3 Arbitrage im Kryptohandel

Die Ursprünge der Arbitrage lassen sich bis ins 14. Jahrhundert in Italien zurückverfolgen, wo Händler den günstigen Erwerb von Wechseln nutzten, um sie an anderen Orten zu höheren Preisen zu veräußern. Dieses Verfahren wurde als „arbitrio“ bezeichnet und verbreitete sich schließlich in Form einer Methode zur Gewinnerzielung durch den Kauf und Verkauf auf unterschiedlichen Märkten in ganz Europa [49]. Es existieren mehrere Arten des Arbitrage-Handels und somit auch verschiedene Definitionen. Im Finanzwesen wird der Begriff Arbitrage, als das Ausnutzen von Preis-, Kurs- und Zinsunterschieden von marktmäßig gehandelten Gütern, insbesondere Aktien, Devisen, Edelmetallen etc., auf verschiedenen Börsen oder Märkten, verstanden [50], [51], [52]. Neben des persönlichen Profits haben Arbitrageure eine wichtige Funktion für die Markttransparenz und Ausgleichsfunktion. Durch den Kauf und Verkauf von Vermögenswerten auf unterschiedlichen Börsen, wird ein Preisausgleich zwischen diesen herbeigeführt [1].

### 2.3.1 Verschiedene Arten von Arbitrage

Es gibt verschiedene Arten von Arbitrage, die sich in ihrer Herangehensweise und den gehandelten Vermögenswerten unterscheiden. Die für den Kryptowährungsmarkt anwendbare Arbitrage-Strategie ist zunächst die Differenzarbitrage. Sie ist die ursprüngliche Art der Arbitrage und bezieht sich auf den gleichzeitigen Kauf und Verkauf von Wertpapieren an verschiedenen Börsen [49].

Die Devisenarbitrage ist eine spezifische Form der Differenzarbitrage. Der Devisenmarkt gilt in der Ökonomie als Vorzeigebispiel für vollständige Konkurrenz [1]. Die gehandelten Währungen sind völlig homogen und es herrscht eine sehr hohe Transparenz der Marktdaten. Zudem ist die Anzahl der Marktteilnehmer sehr groß. Unter diesen Marktbedingungen kann es nur einen Kurs für jede Währung geben. Wenn beispielsweise ein Euro bei Bank A in London zu einem niedrigeren Briefkurs gehandelt wird als bei Bank B in Frankfurt, die bereit ist, einen höheren Geldkurs zu zahlen, werden Händler sofort versuchen, von dieser Diskrepanz zu profitieren [1]. In London wird die gestiegene Nachfrage den Euro-Kurs rasch anheben, während das erhöhte Angebot in Frankfurt dazu führt, dass der Kurs gedrückt wird. Eine solche Differenzarbitrage, auch bekannt als Platzarbitrage, ist risikofrei, da Devisenhändler versuchen, die Geschäfte möglichst gleichzeitig abzuschließen [1].

Eine alternative Methode zur Ausnutzung von Kursdifferenzen besteht in der Verwendung von Drittwährungen, die als Dreiecksarbitrage bezeichnet wird. Dieser Ansatz gewährleistet, dass die sogenannten Kreuzkurse (Umrechnungskurse über Drittwährungen) der Währungen, nicht von den bilateralen Wechselkursen abweichen. Allerdings erfordert dies einen freien internationalen Devisenhandel und die Konvertibilität der Währungen [53].

### 2.3.2 Herausforderungen und Risiken

Auch wenn das Arbitrage-Trading zunächst attraktiv erscheinen mag, gibt es eine diverse Risiken und Herausforderungen, die es zu berücksichtigen gilt:

Die Volatilität von Kryptowährungen ist für den Handel zunächst einmal positiv, da in diesem Fall größere Profite möglich werden. Allerdings besteht das Risiko, bei großen Kursschwankungen auch hohe Verluste zu erleiden, wenn ein ungünstiger Zeitpunkt vorliegt. Angenommen am 07.11.2021 zum Allzeithoch des Bitcoins wurde auf einer Börse zum günstigsten Ask-Kurs gekauft, um auf einer anderen Börse zu einem potenziell gewinnbringenden Bid-Kurs zu verkaufen. Ist dieser beispielhafte Handel aus verschiedenen Gründen nicht zustande gekommen, so hat man bis zum heutigen Tag einen Verlust von etwa 70 % gemacht [54].

Weiterhin gibt es Ausführungsrisiken, die sowohl technischen Schwierigkeiten, wie Serverausfälle oder Verbindungsproblemen, als auch mangelnder Liquidität auf den Börsen

geschuldet sein können [55]. So können Verzögerungen in der Verarbeitung von Aufträgen seitens des Benutzers, aber auch der Börsen entstehen. Des Weiteren sind Gebühren eine Herausforderung, die man in den meisten Fällen nur schwer beeinflussen kann. Vor allem bei Trades mit größeren Volumina können börsenspezifische Gebühren Gewinne deutlich schmälern.

Zusätzlich zu den aufgeführten Risiken, kann es dazu kommen, dass eine Börse ausfällt, betrügerisch handelt oder insolvent wird und der Anleger einen Totalverlust hinnehmen muss. Weitere Risiken und Herausforderungen, die der Börsen geschuldet sind, wurden in Kapitel 2.2 beschrieben.

## 2.4 Historische Tick-Daten

Historische Tick-Daten sind eine Sammlung von Marktdaten, die aus einer zeitlich lückenlosen Abfolge von Preis- und Handelsvolumenänderungen bestehen. Diese Daten werden in Form von Zeitstempeln, Preisen und Volumina für jeden einzelnen Handel aufgezeichnet und bieten detaillierte Informationen über die Marktdynamik und das Handelsverhalten [56].

Die Signifikanz von historischen Tick-Daten liegt in ihrer Fähigkeit, den Nutzern eine granulare Sicht auf die Marktbewegungen zu bieten. Im Gegensatz zu aggregierten Zeitreihendaten, die in der Regel stündliche, tägliche oder wöchentliche Preis- und Volumenkonsolidierungen bieten, liefern Tick-Daten Informationen über jeden einzelnen Handel [57]. Dies ermöglicht es Handelsmuster, Liquidität und Volatilität mit höherer Präzision zu analysieren. Des Weiteren können durch diese Daten Handelsstrategien, Risikomanagementmaßnahmen oder Portfoliooptimierungen entwickelt werden [58].

Ein wichtiger Aspekt bei der Verwendung von historischen Tick-Daten besteht in der Datenqualität [59]. Anbieter hochwertiger Daten wie Tardis verlangen hohe Preise [60]. Da insbesondere Kryptowährungsmärkte von Natur aus unreguliert und fragmentiert sind, können die Tick-Daten von verschiedenen Börsen und Handelsplattformen variieren. Daher ist es entscheidend, unsaubere Datensätze sorgfältig zu bereinigen, um Duplikate, fehlende Daten und Inkonsistenzen zu beseitigen. Darüber hinaus ist es wichtig, den Datenanbieter sorgfältig auszuwählen, um sicherzustellen, dass die bereitgestellten Daten von hoher Qualität und Zuverlässigkeit sind [61].

### 2.4.1 Relationale Datenbanken – PostgreSQL

Relationale Datenbanken, wie z. B. PostgreSQL, bieten im Vergleich zu Analysetools wie Excel oder Numbers, NoSQL-Datenbanken und Python-Bibliotheken wie Pandas entscheidende Vorteile für die Verwaltung und Analyse großer Datenmengen, wie sie bei der Untersuchung von großen Sätzen an historischer Tick-Daten anfallen.

Der wichtigste Vorteil von PostgreSQL gegenüber anderer Programme und Werkzeuge ist die Skalierbarkeit. Während Excel oder Numbers bei großen Datensätzen an ihre Grenzen stoßen und Pandas auf die Größe des Arbeitsspeichers beschränkt ist [62], ist PostgreSQL in der Lage, große Datensätze effizient zu verwalten, da diese auf der Festplatte gespeichert werden [63]. Die eingeschränkte Anzahl von Zeilen in Excel und Numbers kann zu einer verminderten Leistungsfähigkeit oder im gegebenen Anwendungsfall zum Absturz der Anwendung führen. PostgreSQL hingegen bietet selbst bei Millionen von Datensätzen eine hohe Leistungsfähigkeit.

Darüber hinaus ermöglicht PostgreSQL das Durchführen von komplexen Abfragen mithilfe von Structured Query Language (SQL), um Tabellen effizient zusammenzuführen, Arbitrage-Möglichkeiten aufzudecken und potenzielle Profite aufzuzeigen. SQL bietet leistungsstarke Möglichkeiten, um Daten aus mehreren Tabellen zu kombinieren, zu filtern und zu analysieren.

Des Weiteren eignet sich PostgreSQL insbesondere für strukturierte Daten. Es bietet jedoch auch Unterstützung für semistrukturierte Daten, wie JSON [64], sodass die Vorteile eines relationalen Datenbanksystems genutzt und gleichzeitig flexibel auf unterschiedliche Datentypen reagiert werden können [65]. Im Gegensatz dazu sind NoSQL-Datenbanken zwar flexibler in Bezug auf Datenschemas, jedoch weniger leistungsfähig, wenn es um komplexe Abfragen und Joins geht [66], die im gegebenen Anwendungsfall erforderlich sind.

Zusammenfassend zeigt sich, dass PostgreSQL die beste Wahl für die Verwaltung und Analyse großer, strukturierter Datenmengen darstellt. Im Vergleich zu Excel, Numbers, NoSQL-Datenbanken und Python-Bibliotheken wie Pandas bietet PostgreSQL das beste Gesamtpaket aus Skalierbarkeit, Abfrageflexibilität und Unterstützung für komplexe Analysen, die für die Untersuchung historischer Tick-Daten und das Aufdecken von Arbitrage-Möglichkeiten erforderlich sind. Daher empfiehlt es sich, für die Analyse großer Datenmengen und die Erstellung komplexer Abfragen auf PostgreSQL zurückzugreifen. Die Verwendung von PostgreSQL trägt dazu bei, die Qualität der Ergebnisse zu erhöhen und ermöglicht eine effizientere Verwaltung und Analyse von Daten, um fundierte Entscheidungen und Erkenntnisse zu liefern.

## 2.5 Verwandte Arbeiten

Makarov und Schoar untersuchten die Handels- und Arbitragemöglichkeiten in Kryptowährungsmärkten und analysierten die Effizienz dieser Märkte. Ihre Ergebnisse legen nahe, dass Kryptowährungsmärkte im Vergleich zu traditionellen Finanzmärkten weniger effizient sind, was größere Arbitrage-Möglichkeiten bietet [67].

Luciano Sphere zeigt in seinem Blog-Artikel das Abrufen historischen Tick-Daten für verschiedene Handelspaare. Dieser Ansatz ist allerdings unbrauchbar, da die genutzte Plattform Cryptocompare, Einschränkungen für den kostenfreien Zugang hat [68]. Ein Zugang, der die Arbitrage-Funktionen freischaltet, kostet 349,99 USD im Monat.

**Cryptohopper** [69] ist ein Trading-Bot, der es Benutzern ermöglicht, ihre Handelsstrategien auf verschiedenen Kryptowährungsbörsen automatisch auszuführen. Der Bot bietet eine Vielzahl von Funktionen, um sowohl Anfängern als auch professionellen Tradern zu helfen, ihre Handelsaktivitäten zu optimieren. Da Cryptohopper Gebühren in Höhe von 83,25 \$ pro Monat erhebt, sind Tests, Auswertungen und mögliche Vergleiche mit dem entwickelten **Proof of Concept (PoC)** nicht möglich.

In einem Blog-Artikel wird oberflächlich das Entwerfen und Bauen eines Trading-Bots beschrieben. Allerdings liegt der Fokus nicht nur auf Arbitrage-Strategien, sondern auch auf anderen Handelsstrategien [70]. Die Informationen sind nicht von großem Wert, da keine präzisen Ideen oder Vorschläge für das Entwerfen einer Arbitrage-Trading Zielarchitektur vermittelt werden oder neue Erkenntnisse liefern, die zum Entwickeln einer Arbitrage-Trading Anwendung hilfreich wären.

Der **AMM Arbitrageur** [71] ist ein Arbitrage-Trading-Bot, der Arbitragemöglichkeiten auf **DEX** handeln soll. Dieses GitHub-Repository wurde am 07.12.2022 archiviert und ist somit nicht aktuell. Außerdem sind Arbitrage-Trades auf **DEX** wegen ihrer dezentralen Natur und ihrer Eigenschaften, die in Kapitel 2.2 erläutert werden, schwer mit Trades auf **CEX** zu vergleichen.

# Kapitel 3

## Anforderungsanalyse

### 3.1 Anforderungsanalyse

#### 3.1.1 Funktionale Anforderungen

**Anforderung 1:** Die Anwendung muss in der Lage sein, auf die APIs von Binance, Bitstamp und Coinbase zuzugreifen und diese zu unterstützen. Sie soll so implementiert werden, dass zukünftige Änderungen hinzufügbarer Börsen ohne umfangreiche Änderungen an der Kernfunktionalität der Anwendung ermöglicht werden. Der Zugriff auf die Börsen APIs soll mit einem erfolgreichen HTTP-Statuscode erfolgen.

**Anforderung 2:** Orderbücher abrufen: Die Anwendung muss in der Lage sein, Orderbücher von den Börsen abzurufen, um das Handelsvolumen und die Preisentwicklung zu analysieren. Die Daten sollen in Echtzeit aktualisiert werden. Die Ausgabe soll die zehn niedrigsten Briefkurse und die zehn höchsten Geldkurse der ausgewählten Orderbücher darstellen.

**Anforderung 3:** Niedrigste Briefkurse berechnen: Die Anwendung muss in der Lage sein, aus allen von der Anwendung unterstützten Börsen den niedrigsten Briefkurs und die entsprechenden Mengen nach Hinzurechnung der Börsengebühr zu berechnen. Die Ausgabe soll den niedrigsten Briefkurs und die entsprechende Menge für jede Börse darstellen.

**Anforderung 4:** Höchste Geldkurse berechnen: Die Anwendung muss in der Lage sein, den höchsten Geldkurs aus allen von der Anwendung unterstützten Börsen und deren jeweiligen Mengen nach Abzug der Börsengebühr zu berechnen. Die Ausgabe soll den höchsten Geldkurs und die entsprechende Menge für jede Börse darstellen.

- Anforderung 5:** Die Anwendung muss in der Lage sein, potenzielle Arbitragemöglichkeiten zu erkennen, indem es den niedrigsten Briefkurs und den höchsten Geldkurs der unterstützten Börsen vergleicht. Die Ausgabe soll die potenziellen Arbitragemöglichkeiten unter Berücksichtigung der Börsengebühr und der Transaktionskosten darstellen.
- Anforderung 6:** Die Anwendung muss in der Lage sein, sensible Benutzerdaten, wie API-Schlüssel oder Passwörter, sicher aufzubewahren, um unbefugten Zugriff oder Diebstahl zu verhindern. Hierfür muss die Anwendung die Daten aus einer Datei im .env-Format abrufen können.
- Anforderung 7:** Systematische Aufzeichnung von Marktbedingungen und Preisen: Die Anwendung sollte in der Lage sein, allgemeine Marktbedingungen in einer Protokolldatei im .log-Format zu speichern, um das Ergebnis der ausgeführten Funktionen zu dokumentieren. Die Protokolldatei sollte Informationen enthalten, wie beispielsweise die verglichenen Börsen mit dem niedrigsten Brief- und dem höchsten Geldkurs, ein digitaler Zeitstempel, die Uhrzeit, sowie die Handelsvolumina, Brief- und Geldkurse inklusive der Börsengebühren der verglichenen Börsen und alle anderen relevanten Informationen, die während der Verarbeitung der Funktionen für die spätere Analyse wichtig ist. Wenn keine Arbitragemöglichkeit gefunden wird, sollte die Anwendung dennoch in der Lage sein, die oben genannten Informationen in der Protokolldatei zu erfassen, sodass der Benutzer den Anwendungsverlauf nachvollziehen kann.
- Anforderung 8:** Arbitragemöglichkeiten protokollieren: Sobald eine Arbitragemöglichkeit erkannt wird, soll die Anwendung alle relevanten Informationen in eine Protokolldatei im .log-Format speichern. Diese Informationen sollen die jeweiligen Börsen mit dem niedrigsten Briefkurs und dem höchsten Geldkurs, ein digitaler Zeitstempel und die Uhrzeit, das Handelsvolumen der verglichenen Börsen, die Brief- und Geldkurse inklusive der Börsengebühren sowie der erwartete Profit enthalten. Die Protokolldatei soll übersichtlich und leicht verständlich sein, damit der Benutzer schnell und einfach auf die ermittelten Arbitragemöglichkeiten zugreifen kann.
- Anforderung 9:** Automatisierte Handelsfunktion: Die Anwendung muss eine automatisierte Handelsfunktion haben, die den Kauf und Verkauf von Kryptowährungen in Echtzeit durchführen kann. Weiterhin muss die Anwendung die Ergebnisse der Trades in einer separaten Datei im .log-Format speichern.

### 3.1.2 Nicht-funktionale Anforderungen

- Anforderung 1:** Einfacher Download: Der Benutzer soll in der Lage sein, die Anwendung vom GitHub-Repository herunterzuladen. Ein Link zum GitHub-Repository soll im Anhang unter [§](#) der Thesis bereitgestellt werden, damit der Benutzer die Anwendung schnell und einfach finden und herunterladen kann.
- Anforderung 2:** Einfache Installation: Die Installation der Anwendung soll einfach und unkompliziert sein. Der Installationsprozess soll automatisch ablaufen und so gestaltet sein, dass der Benutzer keine speziellen technischen Kenntnisse benötigt. Eine schrittweise Anleitung soll bereitgestellt werden, die dem Benutzer dabei hilft, die Anwendung schnell und problemlos zu installieren. Darüber hinaus soll eine Installationsdatei für alle benutzten Bibliotheken bereitgestellt werden, um sicherzustellen, dass alle Abhängigkeiten der Anwendung automatisch installiert werden.
- Anforderung 3:** Zukünftige Updates: Der Benutzer soll in der Lage sein, zukünftige Updates der Anwendung problemlos zu erhalten, indem er die Befehle „git fetch“ und „git pull“ ausführt. Die Anwendung soll auf einem GitHub-Repository gehostet werden und der Link zum Repository unter Anhang [§](#) der Thesis bereitgestellt werden, damit der Benutzer die laufend aktuell gehaltene Version finden und herunterladen kann.
- Anforderung 4:** Verlässlichkeit: Die Anwendung sollte so konzipiert sein, dass das Risiko von Fehlern oder Abstürzen minimiert wird. Sie sollte in der Lage sein, Fehler zuverlässig zu behandeln und auch bei einem Teilausfall weiterzuarbeiten.
- Anforderung 5:** Kompatibilität: Die Anwendung muss mit einer Vielzahl von Betriebssystemen und Geräten kompatibel sein, um sicherzustellen, dass der Benutzer unabhängig von seinem Standort und den von ihm verwendeten Geräten auf die Anwendung zugreifen kann. Die Kompatibilität mit verschiedenen Betriebssystemen und Versionen soll gewährleistet werden, um eine optimale Benutzererfahrung zu gewährleisten.
- Anforderung 6:** Leistung: Die Anwendung soll in der Lage sein, multiple Anfragen pro Sekunde zu verarbeiten, ohne dass es zu nennenswerten Verzögerungen oder Ausfallzeiten kommt. Dies ist besonders wichtig, wenn



Kurse und Orderbücher von mehreren Börsen abgerufen werden und Handelsentscheidungen in Echtzeit getroffen werden.

**Anforderung 7:** Dokumentation: Die Anwendung soll eine umfassende Dokumentation bereitstellen, die alle Aspekte ihrer Funktionalität, Nutzung und Fehlerbehebung abdeckt. Dadurch wird es Benutzern und Entwicklern ermöglicht, die Anwendung effektiv zu nutzen und mögliche Probleme schnell zu lösen.

# Kapitel 4

## Vorgehensweise / Ansatz

Das Vorgehen zur Anforderungsanalyse und zum Design einer Arbitrage-Trading Zielarchitektur für Kryptowährungen basiert auf einer systematischen und methodischen Herangehensweise. Das Vorgehen umfasst die Phasen der Datensammlung, Datenverarbeitung, Analyse der historischen Tick-Daten, Anforderungsdefinition, Design und Implementierung eines **PoC**. Die wichtigsten Schritte und Erkenntnisse werden im Folgenden detailliert erläutert.

### 4.1 Datensammlung

Zunächst wurden historische Daten von verschiedenen Kryptowährungsbörsen gesammelt, um Arbitragemöglichkeiten aus der Vergangenheit zu identifizieren. Umfangreiche Datensätze im .csv-Format wurden durch den Zugriff auf die Bitcoincharts-Schnittstelle [72] mithilfe des eigenentwickelten Python-Skripts „fetch\_historical\_data.py“ [73] heruntergeladen. Da Bitcoincharts keine Daten für Binance sammelt, wurde das Python-Skript „download-trade.py“ des öffentlichen Binance GitHub-Repositorys [74] genutzt. Um dieses Python-Skript auszuführen und die gewünschten historischen Daten herunterzuladen, wurden folgende Parameter verwendet:

```
python3 download-trade.py -t spot -s BTCUSDT -skip-daily 1  
-startDate 2020-01-01 -endDate 2022-11-18
```

### 4.2 Datenverarbeitung

Eine Herausforderung stellte die Verarbeitung der historischen Tick-Daten dar, da herkömmliche Programme wie Microsoft Excel oder macOS Numbers aufgrund der enormen Datenmengen entweder Schwierigkeiten hatten, die Dateien überhaupt zu öffnen oder nicht in der Lage waren, diese vollständig zu verarbeiten. Die Verwendung von Python-Bibliotheken wie Pandas und Plotly führten ebenfalls zu technischen

Schwierigkeiten und zu Programmabstürzen, sodass eine Analyse mit diesen Werkzeugen nicht möglich war. Gängige Editierprogramme wie Microsofts hausgener Texteditor bieten in der Regel keine Analysemöglichkeiten. Dennoch wurden verschiedene Editoren, wie macOS TextEdit, Microsoft Texteditor oder Notepad++ zu Versuchszwecken getestet. Die Editoren konnten die .csv-Dateien nicht verlässlich öffnen. Einzig der Quelltext-Editor Visual Studio war in der Lage, die .csv-Dateien mit sehr großer Verzögerung zu öffnen. Dieser war trotz langer Verzögerungen, insbesondere beim Scrollen und Anpassungen der Dateien, ein essenziell wichtiges Werkzeug, um die .csv-Dateien im weiteren Verlauf manipulieren zu können.

Aufgrund der genannten Schwierigkeiten fiel die Entscheidung auf die Verwendung von relationalen Datenbanken, dessen Vorteile bereits im Kapitel [2.4.1](#) aufgezeigt wurden.

Um die Daten in PostgreSQL einzulesen, mussten die .csv-Dateien zunächst vorbereitet werden. Mithilfe von Visual Studio Code wurde jede Datei manuell um die erste Zeile erweitert, in dem die Attribute „price“, „amount“ und „unix\_time“ hinzugefügt wurden – die .csv-Dateien von Binance haben weitere Attribute, welche für die durchzuführenden Vergleiche irrelevant sind. Die Erweiterung der Dateien um die genannten Attribute ist obligatorisch. Diese dienen innerhalb der PostgreSQL Datenbanken als Tabellenattribute, welche den Preis, die Anzahl und den Zeitpunkt des Handels repräsentieren.

Da pgAdmin 4 [\[75\]](#) als Client genutzt wird, müssen die Tabellen zunächst mit dem pgAdmin 4-SQL-Abfrage-Werkzeug manuell erzeugt werden. Wir beschränken uns im Folgenden auf die Tabellen der drei Kryptowährungsbörsen Coinbase, Bitstamp und Binance. Erklärungen zu den Attributen sind als Inline-Kommentar angegeben. Exemplarisch sieht die Erzeugung der Börsentabellen für Binance wie folgt aus:

```
create table binanceUSD (  
    id SERIAL, /* Primärschlüssel, fortlaufende Nummer */  
    unwichtig VARCHAR(99), /* binance-spezifisch; irrelevante */  
    price NUMERIC, /* gehandelter Preis */  
    amount NUMERIC, /* gehandelte Menge */  
    quote_amount NUMERIC, /* binance-spezifisch; irrelevante */  
    unix_time BIGINT, /* Unix Zeitstempel des Handels */  
    exchange VARCHAR(99) DEFAULT 'binanceUSD' NOT NULL,  
    /* exchange dient für interoperative Abfragen */  
    PRIMARY KEY (id, exchange) /* Festlegen Primärschlüssel */  
)
```

Da pgAdmin 4 als Server und als eigenständiges Benutzerkonto innerhalb des macOS Betriebssystems fungiert, ist der Zugriff auf das Dateisystem, zumindest unter macOS

Big Sur, nicht ohne weiteres möglich. Trotz Anpassung der Berechtigungen und Vergabe des Festplattenvollzugriffs können die Dateien von pgAdmin 4 nicht aufgerufen werden. Als Inspiration dienten zwei Beiträge auf Stackoverflow [76], [77], bei der durch Verschieben der .csv-Dateien in den temporären Ordner unter „/private/tmp/“ dieses Problem umgangen wird. Anschließend können die .csv-Dateien durch Ausführen des Copy-Befehls im pgAdmin4-SQL-Abfrage-Werkzeug eingelesen werden, siehe Anlagen:

Nachdem sich die Daten in den gewünschten Tabellen befinden, müssen sie harmonisiert werden, da die Daten nicht einheitlich strukturiert sind. Änderungen müssen insbesondere in der Binance-Tabelle vorgenommen werden, da die Attribute „quote\_amount“ und „unwichtig“ aus den Quelldaten nicht benötigt werden. Außerdem weist der Zeitstempel in der Binance-Tabelle eine höhere Genauigkeit auf als der Zeitstempel in den anderen beiden Tabellen. Konkret ist der Zeitstempel in der Binance Tabelle in Millisekunden angegeben, während er in den anderen Tabellen in Sekunden angegeben ist. Für diesen Schritt wird zunächst die Binance Tabelle durch Umbenennen gesichert und die benötigten Attribute in die neue Binance Tabelle mithilfe der Insert-Operation übernommen. Aus Kapazitätsgründen wird die gesicherte Tabelle gelöscht, siehe Anhang unter 8

Um die Analyse zu erleichtern, sind in allen Tabellen zwei zusätzliche Spalten erforderlich. Die erste Spalte ist das Attribut „price\_after\_fee“, das den Handelspreis zuzüglich der Börsengebühr repräsentiert und somit den tatsächlichen Preis des Handels unter Berücksichtigung der Gebühren widerspiegelt. Die zweite Spalte „price\_and\_amount“ enthält Informationen über den tatsächlichen Wert des Handels, der durch die Multiplikation von Handelspreis und -volumen ermittelt wird. Sie gibt somit Auskunft darüber, wie viel tatsächlich für einen bestimmten Handel bezahlt wurde. Abschließend müssen die Tabellen mithilfe einer Join-Operation verknüpft werden, um eine neue Tabelle zu erzeugen, die mindestens zwei Tabellen mit demselben Handelszeitpunkt enthält. Diese Verknüpfung ist nun in der Lage, Hinweise auf potenzielle Arbitrage-Möglichkeiten zu geben. Mithilfe der zusätzlichen Spalten können wir anschließend mögliche Gewinne vergleichen und bewerten.

## 4.3 Datenanalyse

Für die Analyse der historischen Daten wurden verschiedene Analysemethoden angewendet, um Arbitragemöglichkeiten zu identifizieren und zu quantifizieren. Dabei wurden sowohl die Preisunterschiede als auch die unterschiedlichen Gebührenstrukturen berücksichtigt. Die Analyse der Daten erfolgte durch die Verwendung einer PostgreSQL-Datenbank und der Erstellung von SQL-Abfragen, um die gewünschten Ergebnisse zu erzielen. Wie eingangs in Kapitel 4.2 erwähnt, war die Analyse mit

anderen Analysewerkzeugen nicht möglich.

Es konnte festgestellt werden, dass die Qualität der Daten nicht ideal ist, somit ist eine präzise Analyse kaum machbar. Dennoch können potenzielle Arbitragemöglichkeiten nachgewiesen werden. Nach Vergleichen der Tabellen konnten für Bitstamp und Binance insgesamt 11.041.662 Trades und für die Börsen Coinbase und Bitstamp insgesamt 6.025.317 potenzielle Arbitragemöglichkeiten notiert werden. Zwischen Binance und Coinbase gibt es keine Schnittpunkte, da die Daten für Binance vom 01.01.2022 bis 01.12.2023 vorliegen und Coinbase letzter Datensatz von Ende 2019 ist. Auch wenn die Analyse dieser Daten nicht sehr ertragreich war, kann festgehalten werden, dass Arbitragemöglichkeiten im Kryptowährungsmarkt keine Seltenheit sind.

Die Ergebnisse der Analyse dienten als Grundlage für die Definition der Anforderungen an die Arbitrage-Trading Zielarchitektur. Dabei wurden verschiedene Aspekte wie die Reaktionszeit, die Skalierbarkeit und die Flexibilität der Architektur berücksichtigt. Basierend auf den Anforderungen wurde ein Design für die Arbitrage-Trading Zielarchitektur entwickelt.

Abschließend wurde ein **PoC** implementiert, um die Umsetzbarkeit und Effektivität der Zielarchitektur zu demonstrieren. Hierbei wurde ein System entwickelt, das die Arbitrage-Trading-Strategie automatisiert und in Echtzeit die Preisunterschiede zwischen den verschiedenen Börsen überwacht und daraufhin entsprechende Trades protokolliert. Die Implementierung erfolgte mithilfe von verschiedenen Python-Bibliotheken, die in Kapitel **5.3** detailliert aufgezeigt werden, und der ausgewählten Architektur, wobei verschiedene Aspekte wie Modularität, Skalierbarkeit und Zuverlässigkeit berücksichtigt wurden. Das **PoC** konnte in der Theorie zeigen, dass die Arbitrage-Trading Strategie auf Basis der Zielarchitektur effektiv und rentabel umgesetzt werden kann.

# Kapitel 5

## Design

### 5.1 Einleitung

Das aufgestellte System zielt darauf ab, Arbitragemöglichkeiten auf Kryptowährungsmärkten zu identifizieren und zu analysieren, um Benutzern die Ausnutzung von Preisunterschieden zwischen mehreren Börsen zu ermöglichen. Hierfür besteht das System aus verschiedenen Komponenten, die miteinander interagieren, um Daten von verschiedenen Kryptowährungsbörsen zu sammeln, zu verarbeiten und zu analysieren.

### 5.2 Technische Details

Die Anwendung wird lokal eingesetzt. Die verwendete Programmiersprache ist Python 3.8.9. Neben den bekannten Vorteilen, die Python als Programmiersprache bietet, wie z. B. der Einfachheit, Lesbarkeit oder Plattformunabhängigkeit, die eine schnelle Entwicklung und einfache Wartung ermöglichen, hat die Sprache eine große Entwicklergemeinschaft, die unter anderem Dokumentationen und Hilfestellungen bereitstellt oder an ständigen Verbesserungen der Sprache und Werkzeuge arbeitet. Das hat zur Folge, dass die Anwendung auch in Zukunft genutzt und weiterentwickelt werden kann. Des Weiteren bietet Python umfangreiche Bibliotheken, die für eine kurze Entwicklungszeit viel Zeit einsparen kann. Im Folgenden werden die genutzten Bibliotheken erläutert:

Für die Datenbeschaffung von den Börsen werden die Bibliotheken `cbpro` [78], `ccxt` [79], `BitstampClient` [80] und die nativen `Representational State Transfer (REST)-Application Programming Interface (API)`-Schnittstellen [81], [82] verwendet. Zusätzlich werden die Standardbibliotheken `„json“`, `„time“`, `„datetime“` und `„python-decouple“` verwendet. Die `json`-Bibliothek dient der Verarbeitung von Daten im JSON-Format, welches bei der Kommunikation mit den `REST-APIs` verwendet wird. Die `time`- und `datetime`-Bibliotheken sind für die Handhabung von Zeitstempeln, Handelszeit-

punkten und die Messung von Ausführungszeiten verantwortlich. Zur Unterstützung des Deploymentprozesses wird die Bibliothek `python-decouple` verwendet, die es ermöglicht, Umgebungsvariablen, welche unter anderem sensible Daten wie API-Schlüssel und Passwörter beinhalten, in einer `.env`-Datei zu speichern und aus dieser zu laden.

Aufgrund der in Kapitel [2.4.1](#) aufgezeigten Vorteile, wurde für die Datenanalyse der historischen Tick-Daten PostgreSQL als Datenbank verwendet. Für die Datenanalyse der historischen Tick-Daten wurde PostgreSQL als Datenbank verwendet. Die Vorteile von PostgreSQL wurden in Kapitel [2.4.1](#) aufgezeigt.

## 5.3 Systemarchitektur

Die zugrunde liegende Architektur der Anwendung ist eine objektorientierte Struktur, die auf modularer Programmierung basiert. Dies ermöglicht eine flexible und erweiterbare Lösung, die es dem System ermöglicht, kontinuierlich um neue Funktionen und Börsen erweitert zu werden, ohne dass es zu umfangreichen Änderungen kommen muss. Die Börsen-Integration ist so gestaltet, dass sie einfach und ohne Beeinträchtigung anderer Funktionen hinzugefügt werden kann, um den sich ständig ändernden Anforderungen des Marktes gerecht zu werden. Dies stellt sicher, dass die Anwendung effizient, skalierbar und wartbar bleibt. Die Architektur wird detailliert erläutert und veranschaulicht:

- **Exchange-Interface:** Eine abstrakte Basisklasse, die als Grundlage für die spezifischen Börsenklassen (Coinbase, Binance, Bitstamp) dient. Sie definiert die erforderlichen Funktionen und Attribute für die Interaktion mit den Börsen, wie das Abrufen von Orderbüchern, Kaufen und Verkaufen von Assets und das Handeln von Gebühren.
- **Spezifische Börsenklassen:** Jede Börse (Coinbase, Binance, Bitstamp) hat ihre eigene Klasse, die von der Exchange-Klasse erbt. Diese Klassen implementieren die börsenspezifischen Methoden, die erforderlich sind, um mit ihren jeweiligen [APIs](#) zu interagieren, die Authentifizierung zu handhaben und die zurückgegebenen Daten zu verarbeiten.
- **TradeManager-Klasse:** Eine zentrale Klasse, die für die Verwaltung mehrerer Börsen, die Aggregation ihrer Daten und die Ausführung der Handelslogik zur Identifizierung von Arbitragemöglichkeiten verantwortlich ist. Sie enthält Methoden zum Abrufen von Orderbüchern, zur Identifikation der niedrigsten [Ask](#)- und höchsten [Bid](#)-Preise (nach Gebühren), zur Ausführung der Handelslogik für alle Börsen und zum Protokollieren.

- Arbitrage-Erkennung: Eine Subkomponente innerhalb des Trademanagers, die die niedrigsten Verkaufspreise und höchsten Kaufpreise unter Berücksichtigung der Börsengebühren über mehrere Börsen hinweg vergleicht. Wenn eine Arbitragemöglichkeit erkannt wird, protokolliert das System den potenziellen Gewinn und andere relevante Details.
- Hauptklasse: Der primäre Einstiegspunkt und Kontrollzentrum für die Anwendung. Sie initialisiert die spezifischen Klassen der Börsen, erstellt eine TradeManager-Instanz mit den Börsen, richtet das Protokollierungssystem ein und führt kontinuierlich die Handelslogik in einer Schleife aus.
- Protokollierung: Ein umfassendes Protokollierungssystem, das in die Architektur integriert ist, um wichtige Daten, Ereignisse und potenzielle Arbitragemöglichkeiten aufzuzeichnen. Es unterstützt bei der Überwachung der Systemleistung, bei der Fehlerbehebung und bei der Analyse der Marktbedingungen.

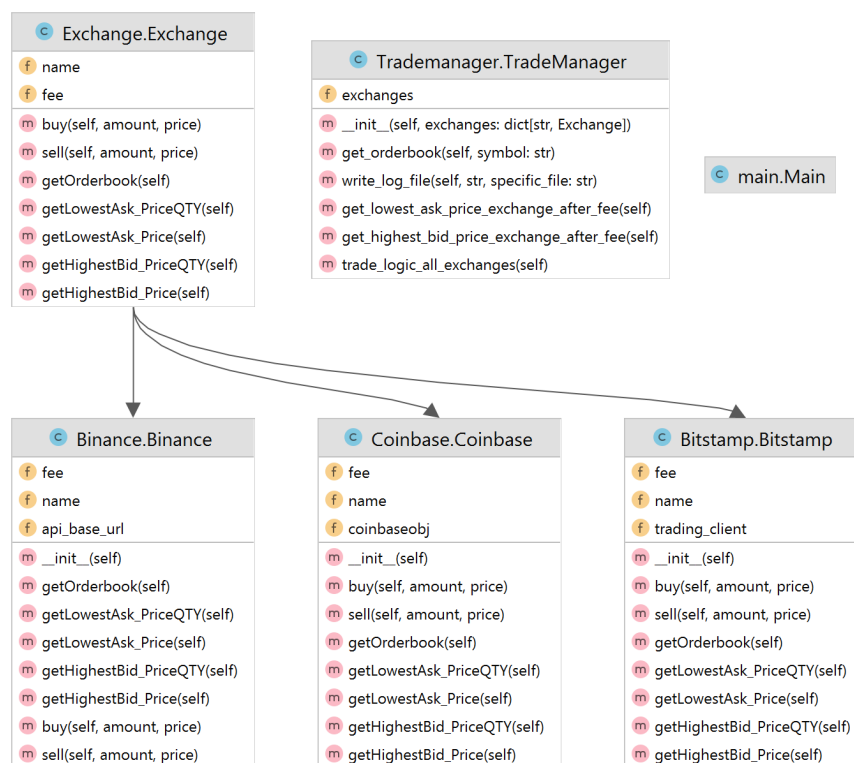


Abbildung 5.1: UML-Klassendiagramm der Architektur



## 5.4 Programmablauf

Das System folgt einem sequentiellen Workflow, um Börsendaten zu sammeln, Arbitragemöglichkeiten zu erkennen und die Ergebnisse zu protokollieren:

1. Initialisierung: Die Hauptklasse initialisiert Instanzen der exchange-spezifischen Klassen (Coinbase, Binance, Bitstamp), richtet das Logging-System ein und übergibt die Exchange-Instanzen an die TradeManager-Klasse.
2. Datensammlung und -verarbeitung: Die TradeManager-Klasse sammelt Order-Book-Daten für die angegebenen Handelspaare (z. B. BTC/USDT) von jeder Börse mithilfe ihrer jeweiligen APIs. Es verarbeitet die zurückgegebenen Daten, um Konsistenz und Nutzbarkeit sicherzustellen.
3. Datenanalyse: Die TradeManager-Klasse berechnet die niedrigsten Angebots- und höchsten Gebotspreise (nach Gebühren) über alle Börsen hinweg. Es vergleicht diese Werte, um potenzielle Arbitrage-Möglichkeiten zu identifizieren.
4. Ausführung der Handelslogik: Die TradeManager-Klasse führt die Handelslogik für alle Börsen aus und prüft auf Arbitrage-Möglichkeiten. Wenn eine Chance gefunden wird, berechnet es den potenziellen Gewinn, zeichnet die Details im Logging-System auf und kann optional im Namen des Benutzers den Handel ausführen. Wenn keine Chance gefunden wird, protokolliert es den aktuellen Marktstatus für zukünftige Analysen.
5. Kontinuierliche Ausführung: Die Hauptklasse führt die Handelslogik in einer Endlosschleife aus, analysiert kontinuierlich den Markt auf Arbitrage-Möglichkeiten. Es überwacht das Logging-System auf Probleme oder bemerkenswerte Ereignisse und kann die Ausführung basierend auf vom Benutzer definierten Kriterien anhalten oder anpassen.
6. Fehlerbehandlung und Wiederherstellung: Das System enthält robuste Fehlerbehandlungsmechanismen, um den kontinuierlichen Betrieb bei unerwarteten Ereignissen wie API-Fehlern, Netzwerkproblemen oder Dateninkonsistenzen zu gewährleisten. Es protokolliert die Fehler und versucht, sich elegant zu erholen, um die Ausfallzeiten zu minimieren und die Zuverlässigkeit des Systems zu gewährleisten.

Durch die Verwendung dieser modularen Architektur, effizienten Arbeitsabläufe und des umfassenden Logging-Systems ist das System in der Lage, Preisunterschiede über mehrere Börsen hinweg in Echtzeit zu identifizieren und auszunutzen.

Das aufgestellte Systemdesign ist auf folgende Wege skalier- und erweiterbar:

- Zusätzliche Börsen können leicht in das System integriert werden, indem die Exchange-Schnittstelle implementiert und die neue Börseninstanz dem Trade Manager hinzugefügt wird.
- Das System kann erweitert werden, um mehrere Handelspaare zu unterstützen, indem Anpassungen zu den gewünschten Handelspaaren in der Hauptklasse gemacht werden können. Implementierungen für Handelspaare sind nicht nötig.
- Der Trade Manager kann weiter optimiert werden, um mehrere Handelspaare und Börsen gleichzeitig zu verwalten, wodurch die Gesamtleistung und Reaktionszeit des Systems verbessert wird. Zusätzlich können so potenziell häufiger Arbitragemöglichkeiten erkannt werden.
- Die Arbitrage Erkennungs-Komponente kann verbessert werden, indem ausgefeiltere Algorithmen und Strategien zur Identifizierung von Arbitragemöglichkeiten implementiert werden, die Faktoren wie Markttiefe, Liquidität und Volatilität berücksichtigen.

### 5.4.1 Systemkomponenten

In diesem Kapitel werden die zentralen Komponenten der Zielarchitektur aufgezeigt, welche ihre spezifischen Rollen und Interaktionen erläutern, um eine effektive Systemfunktionalität zu gewährleisten. Dabei werden deren Funktionen im Zusammenhang mit den erforderlichen Anforderungen der Echtzeitdatenerfassung, der Arbitrage-Analyse und -Ausführung sowie der Fehlerbehandlung detailliert untersucht. Dieser Abschnitt liefert somit eine essenzielle Grundlage für die Durchführung der folgenden Untersuchungen und Analysen im Rahmen dieser Bachelorarbeit.

- **APIs**: **APIs** spielen eine zentrale Rolle im System und ermöglichen die Kommunikation mit verschiedenen Kryptowährungsbörsen. Jede Börse bietet eigene **APIs**, die es Entwicklern ermöglicht, auf ihre Plattform zuzugreifen, Marktdaten zu sammeln und Trades auszuführen. Das System verwendet **REST APIs**, die Standard-HTTP-Methoden (GET, POST, PUT, DELETE) verwenden, um Anfragen zu senden und Antworten zu empfangen. Um eine sichere Kommunikation zu gewährleisten, verwenden die **APIs** **API**-Schlüssel und Geheimschlüssel zur Authentifizierung. Für einfache Interaktionen, z. B. Orderbuchabfragen, wird keine Authentifizierung benötigt.
- **Datenaggregator**: Der Datenaggregator ist eine Komponente innerhalb des Trade-Managers, die für die Sammlung, Verarbeitung und Standardisierung von Daten aus mehreren Börsen verantwortlich ist. Es ruft Orderbuchdaten für **BTC/USDT**

ab und verarbeitet die Daten, um eine Konsistenz zwischen verschiedenen Börsen-APIs sicherzustellen. Diese standardisierten Daten werden dann aus derselben Klasse für die Handelslogik und Ausführung verwendet.

- **Algorithmus-Engine:** Die Algorithmus-Engine ist das Kernstück des Systems und für die Identifizierung und Ausnutzung von Arbitragemöglichkeiten verantwortlich. Sie enthält die Handelslogik, die die durch den Datenaggregator bereitgestellten Marktdaten analysiert, die niedrigsten Ask- und höchsten Bid-Preise über Börsen hinweg findet und potenzielle Gewinne berechnet. Die Algorithmus-Engine speichert derzeit alle Trade-Informationen in einer .log-Datei, auch wenn keine Arbitragemöglichkeiten gefunden werden. Zusätzlich werden Informationen zu gefundenen Arbitragemöglichkeiten in einer separaten .log-Datei gespeichert. Zukünftig ist geplant, dass die Engine dem Benutzer automatisierte Handelsmöglichkeiten bietet und diese Trades in seinem Namen ausführt.
- **Benutzeroberfläche (UI):** Es handelt sich um eine konsolenbasierte Anwendung, die keine gewohnte UI bietet und Anpassungen sowie Erweiterungen ausschließlich programmatisch durchgeführt werden müssen. Der Benutzer hat jedoch die Möglichkeit, auf umfangreiche .log-Dateien zuzugreifen, die detaillierte Informationen bereitstellen.

# Kapitel 6

## Analyse und Test

In diesem Kapitel werden die Test- und Analysemethoden beschrieben, die verwendet werden, um die in Kapitel 3 aufgelisteten Anforderungen zu validieren. Dazu werden verschiedene Testfälle und Szenarien entwickelt, um sicherzustellen, dass die Anwendung alle funktionalen und nicht funktionalen Anforderungen erfüllt.

### 6.1 Funktionale Tests

Für jede der funktionalen Anforderungen werden Testfälle erstellt, um deren Erfüllung zu überprüfen.

#### Test zu Anforderung 1:

Testfall und Beschreibung:

Testen der Verbindung zu den APIs von Binance, Bitstamp und Coinbase.

Testen der Erweiterbarkeit der Anwendung bei Hinzufügen neuer Börsen-APIs

Erwartete Ergebnisse:

- ☒ Beim erfolgreichen Verbinden zu den APIs der Börsen wird bei Binance und Bitstamp der Statuscode 200 erwartet.
- ☒ Testen der Erweiterbarkeit der Anwendung bei Hinzufügen neuer Börsen-APIs.
- ☒ Bei Coinbase wird der Statuscode nur bei Fehlern ausgegeben.

Ergebnisse:

- Statuscode 200 bei Binance und Bitstamp erhalten. Bei Coinbase keine Fehlermeldung.

- Die Börse Coinbase wurde nachträglich als letzte Börse hinzugefügt. Die Anwendung ist ohne große Änderungen erweiterbar.

Fehler oder Anmerkungen: Screenshots

## Test zu Anforderung 2:

Testfall und Beschreibung:

Überprüfen der korrekten Abrufung der Orderbücher von den Börsen.

Testen der Echtzeit-Aktualisierung der Orderbuchdaten.

Überprüfen der korrekten Darstellung der niedrigsten Briefkurse und höchsten Geldkurse

Erwartete Ergebnisse:

- ☒ Es wird erwartet, dass das Orderbuch für jede Börse ausgegeben wird.
- ☒ Es wird erwartet, dass das Orderbuch für jede Börse in Echtzeit aktualisiert wird.
- ☒ Die korrekte Darstellung des niedrigsten Ask- und höchsten Bidkurses wird erwartet.

Ergebnisse:

- Die Ausgabe des Orderbuchs war für jede Börse erfolgreich.

```
[>>> import Bitstamp
{'bids': [[28614.0, 0.03944154], [28611.0, 1.1389344], [28609.0, 0.34913252], [28608.0, 0.21356828], [28607.0, 0.08004267], [28606.0, 0.0395], [28605.0, 0.20565672], [28602.0, 0.38862917], [28599.0, 0.46691603], [28596.0, 1.52057721]], 'asks': [[28623.0, 0.34932789], [28626.0, 0.17400861], [28629.0, 0.0395], [28630.0, 1.60775363], [28631.0, 0.25124712], [28633.0, 0.0395], [28634.0, 0.13972428], [28635.0, 0.34871942], [28637.0, 0.0395], [28639.0, 0.05731057]]}
[>>> import Coinbase
{'bids': [[28621.95, 0.035], [28621.18, 0.12528479], [28621.17, 0.11003925], [28621.1, 0.13544842], [28618.79, 0.035], [28618.45, 0.035], [28617.32, 0.07336944], [28617.31, 0.12228232], [28615.85, 0.035], [28615.59, 0.3278881]], 'asks': [[28631.53, 0.035], [28631.91, 0.036666], [28632.23, 0.08731764], [28632.25, 0.13544842], [28632.65, 0.035], [28632.95, 0.04719865], [28632.96, 0.042], [28633.24, 0.0366643], [28634.5, 0.0366268], [28634.51, 0.035]]}
[>>> import Binance
200
{'bids': [[28628.72, 3.92165], [28628.71, 3.45402], [28628.7, 0.00431], [28628.69, 4.07501], [28628.6, 0.00052], [28628.59, 0.03494], [28628.58, 0.04], [28628.35, 1.387], [28628.3, 0.4], [28628.2, 0.1]], 'asks': [[28628.73, 2.08052], [28628.74, 0.62667], [28628.78, 0.00271], [28628.79, 0.00864], [28628.84, 0.00681], [28628.88, 0.00134], [28628.9, 0.0004], [28629.0, 0.55028], [28629.01, 0.00184], [28629.08, 0.00038]]}
```

Abbildung 6.1: Orderbuchabfrage für Bitstamp, Coinbase und Binance

- Die Orderbücher werden in Echtzeit mithilfe einer Endlosschleife für alle drei Börsen aktualisiert und ausgegeben.

- Der niedrigste **Ask**- und höchste **Bid**kurs wird für jede Börse korrekt dargestellt:

```
[>>> import Binance
Niedrigster Ask-Preis: 28570.0
Hoechster Bid-Preis: 28569.99
[>>> import Bitstamp
Niedrigster Ask-Preis: 28574.0
Hoechster Bid-Preis: 28567.0
[>>> import Coinbase
Niedrigster Ask-Preis: 28576.18
Hoechster Bid-Preis: 28565.44
```

Abbildung 6.2: Niedrigste **Ask**- und höchste **Bid**kurse

Fehler oder Anmerkungen: Aus Gründen der Übersicht wurde auf ein Screenshot für Test 2 verzichtet.

### Test zu Anforderung 3:

Testfall und Beschreibung:

Überprüfen und darstellen der korrekten Berechnung der niedrigsten **Ask**kurse und Mengen unter Berücksichtigung der Börsengebühren für jede Börse.

Erwartete Ergebnisse:

- ☐ Ausgabe des korrekt berechneten **Ask**kurses nach spezifischer Börsengebühr und der vorhandenen Menge.

Ergebnisse:

- Die Berechnung und Darstellung der niedrigsten **Ask**kurse und Menge ist korrekt.

```
[>>> import Binance
Niedrigster Ask-Preis und Menge (exkl. Gebühren): [28546.47, 3.10242]
Niedrigster Ask-Preis und Menge (inkl. Gebühren): [28575.01647, 3.10242]
Hoechster Bid-Preis und Menge (exkl. Gebühren): [28546.46, 0.07365]
Hoechster Bid-Preis und Menge (inkl. Gebühren): [28575.006459999997, 0.07365]
[>>> import Coinbase
Niedrigster Ask-Preis und Menge (exkl. Gebühren): [28550.74, 0.08718036]
Niedrigster Ask-Preis und Menge (inkl. Gebühren): [28722.04444, 0.08718036]
Hoechster Bid-Preis und Menge (exkl. Gebühren): [28539.65, 0.0875623]
Hoechster Bid-Preis und Menge (inkl. Gebühren): [28710.8879, 0.0875623]
[>>> import Bitstamp
Niedrigster Ask-Preis und Menge (exkl. Gebühren): [28554.0, 0.07840996]
Niedrigster Ask-Preis und Menge (inkl. Gebühren): [28554.0, 0.07840996]
Hoechster Bid-Preis und Menge (exkl. Gebühren): [28541.0, 0.35026024]
Hoechster Bid-Preis und Menge (inkl. Gebühren): [28541.0, 0.35026024]
>>> █
```

Abbildung 6.3: **Ask** und **Bid**kurse nach Gebühr und Menge

Fehler oder Anmerkungen: Keine

#### Test zu Anforderung 4:

Testfall und Beschreibung:

Überprüfen und darstellen der korrekten Berechnung der höchsten **Bid**kurse und Mengen unter Berücksichtigung der Börsengebühren für jede Börse.

Erwartete Ergebnisse:

- ☐ Ausgabe des korrekt berechneten **Bid**kurses nach spezifischer Börsengebühr und der vorhandenen Menge.

Ergebnisse:

- Die Berechnung und Darstellung der höchsten **Bid**kurse und Menge ist korrekt.

Fehler oder Anmerkungen: Screenshot des Tests, s. Abbildung **6.3**

#### Test zu Anforderung 5:

Testfall und Beschreibung:

Überprüfen der korrekten Identifizierung von Arbitragemöglichkeiten durch den Vergleich der niedrigsten **Ask**- und höchsten **Bid**kurse der unterstützten Börsen.

Testen der Berücksichtigung von Börsengebühren bei der Darstellung potenzieller Arbitragemöglichkeiten.

Erwartete Ergebnisse:

- ☒ Es wird erwartet, dass Arbitragemöglichkeiten erkannt werden und diese unter Berücksichtigung der Börsengebühr dargestellt werden.

Ergebnisse:

- Arbitragemöglichkeiten werden erkannt und die Gebühren zum **Ask**- und **Bid**preis je nach Börse berücksichtigt.

```

Arbitrage Moeglichkeit !!!
Timestamp: 1678316171.829556 2023-03-08 23:56:11.829556
Ask und QTY bei Binance: [21751.99, 0.01089] und Bid bei Bitstamp: [21783.0, 0.51]
Nach Gebuehren liegt der gewuenschte Verkaufspreis bei Bitstamp ueber 21773.74199 Aktuell:
21783.0 und somit existiert eine Arbitrage Moeglichkeit

Arbitrage Moeglichkeit !!!
Timestamp: 1678316172.988557 2023-03-08 23:56:12.988557
Ask und QTY bei Binance: [21729.41, 0.0217] und Bid bei Bitstamp: [21783.0, 0.51]
Nach Gebuehren liegt der gewuenschte Verkaufspreis bei Bitstamp ueber 21751.139409999996
Aktuell: 21783.0 und somit existiert eine Arbitrage Moeglichkeit

Timestamp: 1678316174.187432 2023-03-08 23:56:14.187432
Ask und Anzahl bei Binance: [21725.47, 0.02785]
Nach Gebuehren muss der gewuenschte Verkaufspreis bei Bitstamp ueber 21747.19547 liegen.
Aktuell: [21724.0, 0.00629746]

Timestamp: 1678316175.188385 2023-03-08 23:56:15.188385
Ask und Anzahl bei Binance: [21715.15, 0.12689]
Nach Gebuehren muss der gewuenschte Verkaufspreis bei Bitstamp ueber 21736.865149999999
liegen. Aktuell: [21724.0, 0.00629746]

```

Abbildung 6.4: Identifizierung von Arbitragemöglichkeiten

Fehler oder Anmerkungen: Keine Coinbase Ausgaben, da Gebühren zu hoch sind und die Handelslogik sich für Binance und Bitstamp entscheidet und diese protokolliert.

## Test zu Anforderung 6:

Testfall und Beschreibung:

Testen der sicheren Speicherung und Verwaltung von Benutzerdaten wie API-Schlüsseln und Passwörtern.

Erwartete Ergebnisse:

- ☒ Zugriff auf die **APIs** muss mit der .env-Datei möglich sein, ohne dass sensible Daten im Quellcode als Klartext hinterlegt sind.

Ergebnisse:

- Zugriff auf Passwörter durch .env-Datei ist erfolgreich. Siehe Bitstamp-Klasse Zeile 16:

```

self.trading_client =
bitstamp.client.Trading(username=config('username_bitstamp',
default=''), key=config('api_key_bitstamp', default=''),
secret=config('api_secret_bitstamp', default=''))

```

Fehler oder Anmerkungen: Für einfache **API**-Zugriffe werden für Coinbase und Binance keine sensiblen API-Daten benötigt.

## Test zu Anforderung 7:

Testfall und Beschreibung:



Überprüfen der systematischen Aufzeichnung von Marktbedingungen und Preisen in einer Protokolldatei im .log-Format.

Testen der Protokollierung von relevanten Informationen, auch wenn keine Arbitragemöglichkeiten gefunden wurden.

Erwartete Ergebnisse:

- ☒ Erfolgreiche Protokollierung einer Datei im .log-Format, die die Informationen Zeitpunkt, **Ask**- und **Bid**preise nach Gebühren und die jeweils verglichenen Börsen angibt. Zusätzlich wird erwartet, dass der Hinweis „Keine Arbitragemöglichkeit gefunden“ in der Protokolldatei festgehalten wird.

Ergebnisse:

- .log-Datei wird erfolgreich protokolliert. Die gewünschten Informationen sind enthalten.

```
Timestamp: 1679314928.541204 2023-03-20 13:22:08.541204
Niedrigster Ask Preis und QTY bei: Bitstamp 28160.0 und 0.26547169
Hoechster Bid Preis und QTY bei: Binance 28148.95287 und 0.03375
obige Preise sind nach Gebuehren, somit keine Arbitrage Moeglichkeit!

Timestamp: 1679314929.625421 2023-03-20 13:22:09.625421
Niedrigster Ask Preis und QTY bei: Bitstamp 28160.0 und 0.26547169
Hoechster Bid Preis und QTY bei: Binance 28147.20462 und 0.04131
obige Preise sind nach Gebuehren, somit keine Arbitrage Moeglichkeit!

Timestamp: 1679314987.742232 2023-03-20 13:23:07.742232
Niedrigster Ask Preis und QTY bei: Bitstamp 28152.0 und 0.0305
Hoechster Bid Preis und QTY bei: Binance 28147.57425 und 0.004
obige Preise sind nach Gebuehren, somit keine Arbitrage Moeglichkeit!

Timestamp: 1679315025.5659058 2023-03-20 13:23:45.565906
Niedrigster Ask Preis und QTY bei: Bitstamp 28188.0 und 0.03536118
Hoechster Bid Preis und QTY bei: Binance 28166.175629999998 und 0.035
obige Preise sind nach Gebuehren, somit keine Arbitrage Moeglichkeit!
```

Abbildung 6.5: .log-Datei ohne Arbitragemöglichkeiten

Fehler oder Anmerkungen: Keine

## Test zu Anforderung 8:

Testfall und Beschreibung:

Überprüfen der Protokollierung erkannter Arbitragemöglichkeiten und der Speicherung relevanter Informationen in einer Protokolldatei im .log-Format.

Erwartete Ergebnisse:

- Erfolgreiche Protokollierung einer Datei im .log-Format, die die Informationen: Zeitpunkt der Arbitragemöglichkeit, **Ask**- und **Bid**preise nach Gebühren, die für die Arbitragemöglichkeit verglichenen Börsen und **Return of Trade (RoT)** enthält. Zusätzlich wird der Hinweis, dass eine Arbitragemöglichkeit gefunden wurde, in der Protokolldatei erwartet.

Ergebnisse:

- ☒ .log-Datei wird erfolgreich protokolliert. Die gewünschten Informationen sind enthalten.

```
Arbitrage Moeglichkeit !!!  
Timestamp: 1679311943.9164102 2023-03-20 12:32:23.916410  
Niedrigster Ask Preis und QTY bei: Bitstamp 28200.0 und 9.6767616  
Hoechster Bid Preis und QTY bei: Binance 28203.258510000003 und 0.01483  
Obige Preise sind nach Gebuehren und somit existiert eine Arbitrage Moeglichkeit  
Return of Trade wenn erfolgreich ausgefuehrt waere: 0.04832370330007052  
  
Arbitrage Moeglichkeit !!!  
Timestamp: 1679311944.8137238 2023-03-20 12:32:24.813724  
Niedrigster Ask Preis und QTY bei: Bitstamp 28200.0 und 9.6767616  
Hoechster Bid Preis und QTY bei: Binance 28200.771 und 0.054  
Obige Preise sind nach Gebuehren und somit existiert eine Arbitrage Moeglichkeit  
Return of Trade wenn erfolgreich ausgefuehrt waere: 0.04163400000015827  
  
Arbitrage Moeglichkeit !!!  
Timestamp: 1679313286.784808 2023-03-20 12:54:46.784808  
Niedrigster Ask Preis und QTY bei: Bitstamp 28158.0 und 0.26547545  
Hoechster Bid Preis und QTY bei: Binance 28159.572239999998 und 0.95191  
Obige Preise sind nach Gebuehren und somit existiert eine Arbitrage Moeglichkeit  
Return of Trade wenn erfolgreich ausgefuehrt waere: 0.417391121507535
```

Abbildung 6.6: .log-Datei, die Arbitragemöglichkeiten protokolliert

Fehler oder Anmerkungen: Keine

### Test zu Anforderung 9:

Testfall und Beschreibung:

Überprüfen der Funktionsfähigkeit der automatisierten Handelsfunktion, um Kauf- und Verkaufstransaktionen von Kryptowährungen in Echtzeit durchzuführen.

Testen der Speicherung der Ergebnisse der Trades in einer separaten Datei im .log-Format.

Erwartete Ergebnisse:

- Es wird erwartet, dass die Anwendung automatisiert handelt und Profite erzielt.
- Es wird erwartet, dass die Trades in einer .log-Datei protokolliert werden.

Ergebnisse:

- Keine Ergebnisse, da Handelsfunktion nicht implementiert wurde.

Fehler oder Anmerkungen: Anforderung nicht erfüllt, da Handelsfunktion nicht implementiert wurde.

## 6.2 Nicht-funktionale Tests

### Test zu Anforderung 1:

Testfall und Beschreibung:

- ☐ Überprüfen des Download-Links im Anhang [8](#) und Download der Anwendung aus dem GitHub-Repository.

Erwartete Ergebnisse:

- ☒ Es wird erwartet, dass der Link zum gewünschten GitHub-Repository führt.
- ☒ Es wird erwartet, dass der Download mit „git clone“ ohne Probleme möglich ist.

Ergebnisse:

- Die Anwendung ist über den Link unter [73](#) erreichbar.
- Der Download der Anwendung funktioniert problemlos

```
PS C:\Users\...\Downloads\Pyprojekte> git clone https://github.com/ebugday81/arbitrabort
Cloning into 'arbitrabort'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 3), reused 11 (delta 3), pack-reused 0
Unpacking objects: 100% (11/11), done.
PS C:\Users\...\Downloads\Pyprojekte>
```

Abbildung 6.7: Ausgabe nach erfolgreichem Klonen des GitHub-Repositorys

Fehler oder Anmerkungen: Benutzername im Screenshot entfernt.

### Test zu Anforderung 2:

Testfall und Beschreibung:

- ☐ Überprüfen, ob die Durchführung der Installation ohne weitere Kenntnisse funktioniert und ob alle benötigten Bibliotheken und Abhängigkeiten mithilfe der Anleitung erfolgreich installiert werden.

Erwartete Ergebnisse:

- ☐ Es wird erwartet, dass die Anwendung ohne komplizierte Benutzerinteraktion installiert wird.

- Es wird erwartet, dass alle benötigten Bibliotheken und Abhängigkeiten mit Hilfe der Installationsdatei installiert werden, sodass die Anwendung nach erfolgreicher Installation startbereit ist.

Tatsächlich erhaltene Ergebnisse:

- Die Installation der Anwendung funktioniert problemlos. Mit einer modernen IDE wie PyCharm [83] wird die **requirements.txt** automatisch ausgeführt und alle Bibliotheken und Abhängigkeiten automatisch installiert.

Fehler oder Anmerkungen: Keine Screenshots, da Installation automatisch erfolgt ist.

### Test zu Anforderung 3:

Testfall und Beschreibung:

- Überprüfen, ob die Aktualisierung der Anwendung und Durchführung eines Updates nach Anleitung funktioniert.

Erwartete Ergebnisse:

- Es wird erwartet, dass die Aktualisierung der Anwendung und Durchführung eines Updates mittels „git fetch“ und „git pull“ aktualisiert wird.

Ergebnisse:

- Die Aktualisierung der Anwendung und Durchführung des Updates ist problemlos möglich.

```
[          :arbitrabort Bugday$ git fetch
[          :arbitrabort Bugday$ git pull
Updating 2a94487..6479c24
Fast-forward
  Binance.py      | 18 ++++++-----
  Bitstamp.py     | 34 ++++++-----
  Coinbase.py     | 18 ++++++-----
  Trademanager.py |  2 +-
  main.py         | 16 ++++++-----
  5 files changed, 43 insertions(+), 45 deletions(-)
DerChef-MB:arbitrabort Bugday$ █
```

Abbildung 6.8: Ausgabe nach erfolgreichem Update

Fehler oder Anmerkungen: Keine

## Test zu Anforderung 4:

Testfall und Beschreibung:

- Überprüfen, ob durch die Simulation eines Teilausfalls oder Fehlern die Anwendung weiterhin eingeschränkt lauffähig ist und keine Abstürze verursacht.

Erwartete Ergebnisse:

- Die Anwendung ist nach der Entfernung der Coinbase-Instanz aus der zu verarbeitenden Liste weiterhin eingeschränkt lauffähig. Die übrigen Börsen werden weiterhin kontinuierlich verglichen und das Programm stürzt nicht ab.
- Die Anwendung läuft trotz Verbindungsfehler einer Börse weiter und vergleicht die übrigen Börsen kontinuierlich weiter.

Ergebnisse:

- Nach Entfernung der Coinbase-Instanz aus der Liste der Börsen lief die Anwendung problemlos weiter.

```
if __name__ == '__main__':  
    coinbase = Coinbase()  
    bitstamp = Bitstamp()  
    binance = Binance()  
    list_of_exchanges = {'Binance': binance,  
                        'Bitstamp': bitstamp}#, #'Coinbase': coinbase}
```

Abbildung 6.9: Quelltextänderung, um Coinbase-Teilausfall zu simulieren

```
Anwendung wird am: 2023-03-31 09:27:30.351796 auf folgendem System gestartet:  
Windows  
Schleifendurchlauf: #1  
Bidex: (27769.34286, <Binance.Binance object at 0x034C1670>, 0.54846) und Askex: (27808.0, <Bitstamp.Bitstamp object at 0x005F7718>, 0.29)  
Schleifendurchlauf: #2  
Bidex: (27769.34286, <Binance.Binance object at 0x034C1670>, 0.69846) und Askex: (27808.0, <Bitstamp.Bitstamp object at 0x005F7718>, 0.29)  
Schleifendurchlauf: #3  
Bidex: (27769.34286, <Binance.Binance object at 0x034C1670>, 0.70224) und Askex: (27808.0, <Bitstamp.Bitstamp object at 0x005F7718>, 0.29)  
Schleifendurchlauf: #4  
Bidex: (27767.18502, <Binance.Binance object at 0x034C1670>, 0.00038) und Askex: (27808.0, <Bitstamp.Bitstamp object at 0x005F7718>, 0.29)  
Schleifendurchlauf: #5
```

Abbildung 6.10: Ausgabe nach Simulation des Coinbase-Teilausfalls

- Nach synthetisch erzeugten Verbindungsfehlern für die Coinbase-Instanz, läuft das Programm weiter. Die übrigen Börsen werden weiterhin kontinuierlich verglichen.

```

Anwendung wird am: 2023-03-31 10:24:43.366019 auf folgendem System gestartet:
Windows
Schleifendurchlauf: #1
{'Binance': <Binance.Binance object at 0x034E7280>, 'Bitstamp': <Bitstamp.Bitstamp object at 0x0349EA18>, 'Coinbase': <Coinbase.Coinbase object at 0x002A7718>}
Binance
Binance mit [27768.08, 5.96287] und mit Gebühr: 27795.84808
Bitstamp
Bitstamp mit [27773.0, 0.3599142] und mit Gebühr: 27773.0
Coinbase
'NoneType' object is not subscriptable
'NoneType' object is not subscriptable
Bisex: (27740.30193, <Binance.Binance object at 0x034E7280>, 1.16316) und Askex: (27773.0, <Bitstamp.Bitstamp object at 0x0349EA18>, 0.3599142)
Schleifendurchlauf: #2
{'Binance': <Binance.Binance object at 0x034E7280>, 'Bitstamp': <Bitstamp.Bitstamp object at 0x0349EA18>, 'Coinbase': <Coinbase.Coinbase object at 0x002A7718>}
Binance
Binance mit [27768.08, 8.19634] und mit Gebühr: 27795.84808
Bitstamp
Bitstamp mit [27773.0, 0.3599142] und mit Gebühr: 27773.0
Coinbase

```

Abbildung 6.11: Erfolgreiche Fehlerbehandlung eines Börsenausfalls

Fehler oder Anmerkungen: **API**-Fehlern, die allerdings in Vergangenheit souverän abgefangen wurden.

## Test zu Anforderung 5:

Testfall und Beschreibung:

- ☐ Überprüfen, ob die Anwendung auf verschiedenen Betriebssystemen und Endgeräten lauffähig ist.

Erwartete Ergebnisse:

- ☒ Die Anwendung läuft auf den in Kapitel **6.3** aufgeführten Endgeräten.

Ergebnisse:

- Die Anwendung ist auf beiden Endgeräten mit unterschiedlichen Betriebssystemen lauffähig, siehe **6.12** und **6.13**

```

PS C:\Users\Der Chef\Downloads\Pyprojekte\arbitrabot> python .\main.py
Anwendung wird am: 2023-03-31 05:48:38.517448 auf folgendem System gestartet:
Windows
Schleifendurchlauf: #1
Schleifendurchlauf: #2
Schleifendurchlauf: #3
Schleifendurchlauf: #4
Schleifendurchlauf: #5
Schleifendurchlauf: #6
Schleifendurchlauf: #7

```

Abbildung 6.12: Ausgabe der Lauffähigkeit auf Windows

```
[ arbitrabot Bugday$ python3 main.py
Anwendung wird am: 2023-03-31 06:49:46.440526 auf folgendem System gestartet:
Darwin
Schleifendurchlauf: #1
Schleifendurchlauf: #2
Schleifendurchlauf: #3
Schleifendurchlauf: #4
Schleifendurchlauf: #5
Schleifendurchlauf: #6
Schleifendurchlauf: #7
Schleifendurchlauf: #8
```

Abbildung 6.13: Ausgabe der Lauffähigkeit auf macOS

Fehler oder Anmerkungen: Darwin ist der Betriebssystemkern von macOS.

### Test zu Anforderung 6:

Testfall und Beschreibung:

- ☐ Überprüfen, ob die Anwendung imstande ist, mehrere Anfragen pro Sekunde auszuführen.

Erwartete Ergebnisse:

- ☒ Es wird erwartet, dass die Anwendung Anfragen ohne nennenswerte Verzögerungen oder Ausfälle verarbeitet.

Ergebnisse:

- Die Anwendung ist in der Lage mehrere Anfragen pro Sekunde zu verarbeiten.

Fehler oder Anmerkungen: Latenzen der Börsen beeinflussen das Ergebnis. So dauert jeder Programmdurchlauf zwischen 0.8 - 1,2 Sekunden. Für weitere Details, siehe .log-Datei im Anhang unter [8](#).

### Test zu Anforderung 7:

Testfall und Beschreibung:

- ☐ Überprüfen, ob die Dokumentation vollständig und verständlich ist.

Erwartete Ergebnisse:

- ☐ Es wird erwartet, dass die Dokumentation die Funktionalität, Nutzung und Fehlerbehebung abdeckt und klar, präzise und leicht verständlich ist.

Ergebnisse:

- Die Dokumentation ist zum Zeitpunkt des 31.03.2023 unvollständig.

Fehler oder Anmerkungen: Aufgrund anderer Prioritäten wurde die Dokumentation nach hinten verschoben.

## 6.3 Testumgebung

Um ein umfassendes Verständnis der Leistungsfähigkeit und Kompatibilität der Anwendung zu gewährleisten, wurden während der Tests zwei verschiedene Endgeräte mit unterschiedlichen Betriebssystemen eingesetzt. Dies war notwendig, um die Erfüllung der nicht-funktionalen Anforderungen zu überprüfen und sicherzustellen, dass das Produkt unabhängig vom verwendeten Gerät oder Betriebssystem einwandfrei funktioniert.

Endgerät 1 - zusammengestellter Desktop PC:

- Prozessor: Intel i5-3570k @ 3.40GHz
- Arbeitsspeicher: 8,00 GB 1600 MHz DDR3
- Betriebssystem: macOS Big Sur V11.7
- Pythonversion: 3.8.3

Endgerät 2 - MacBook Pro Mitte 14:

- Prozessor: 2,6 GHz Dual-Core Intel Core i5
- Arbeitsspeicher: 8,00 GB 1600 MHz DDR3
- Betriebssystem: Microsoft Windows 7 Professional N SP1
- Pythonversion: 3.8.12



# Kapitel 7

## Bewertung

### 7.1 Bewertung und Fazit

#### 7.1.1 Bewertung

Die entwickelte Anwendung für die Arbitrage-Trading Zielarchitektur hat den Großteil ihrer Ziele erreicht. Sie kann Orderbücher von drei der größten Kryptowährungsbörsen (Binance, Bitstamp und Coinbase) anzeigen und die Geld- und Briefkurse vergleichen. Dies ermöglicht es, Arbitragemöglichkeiten in Echtzeit zu erkennen und zu protokollieren.

Die Anwendung hat mehrere Stärken:

1. Echtzeit-Orderbuchdaten: Durch die Nutzung der APIs der drei Börsen ist die Anwendung in der Lage, Echtzeit-Orderbuchdaten zu erhalten und diese für den Benutzer aufzubereiten.
2. Automatische Erkennung von Arbitragemöglichkeiten: Die Anwendung erkennt Arbitragemöglichkeiten automatisch und protokolliert sie für den Benutzer. Dies ermöglicht es, schnell auf Marktineffizienzen zu reagieren und potenzielle Gewinne zu erzielen.
3. Flexibilität: Die Anwendung kann leicht auf zusätzliche Kryptowährungsbörsen erweitert werden, um die Chancen auf erfolgreiche Arbitragegeschäfte zu erhöhen.

Trotz dieser Stärken gibt es einige Einschränkungen und Verbesserungspotenziale:

1. Latenz: Die Anwendung ist auf die Latenz der APIs der Börsen angewiesen, was zu Verzögerungen bei der Erkennung von Arbitragemöglichkeiten führen kann. Eine mögliche Lösung könnte die Nutzung von Websockets oder anderen Technologien sein, um die Latenz weiter zu reduzieren.

2. Automatisierung des Handels: Die Anwendung erkennt und protokolliert zwar Arbitragemöglichkeiten, führt jedoch keine automatisierten Handelsaktionen durch. Eine Erweiterung der Anwendung, um automatisierten Handel zu ermöglichen, könnte den Benutzern helfen, noch schneller auf Arbitragechancen zu reagieren.

### **7.1.2 Fazit**

Insgesamt bietet die Bachelorthesis eine solide Grundlage für zukünftige Weiterentwicklungen im Bereich der Arbitrage-Trading Automatisierungen. Durch die Integration der vorgeschlagenen Verbesserungen unter [7.2](#) könnten die Rentabilität und Effektivität der Anwendung in der Zukunft gesteigert werden. Gleichzeitig ist es entscheidend, die Risiken und regulatorischen Anforderungen bei der Auswahl und Nutzung von Kryptowährungsbörsen sorgfältig zu prüfen, um langfristig erfolgreiche Arbitragegeschäfte durchführen zu können.

## **7.2 Ausblick**

In Anbetracht der in dieser Thesis identifizierten Verbesserungspotenziale und Einschränkungen gibt es mehrere Möglichkeiten, die Anwendung in der Zukunft weiterzuentwickeln und zu optimieren, welche nachfolgend aufgeführt werden:

### **7.2.1 Verbesserung der Handelslogik**

Die aktuelle Handelslogik der Anwendung könnte verbessert werden, indem fortschrittlichere Algorithmen und Strategien implementiert werden. Dies könnte dazu beitragen, die Rentabilität von Arbitragegeschäften zu erhöhen und das Risiko zu verringern. Parallelisierungen könnten ebenfalls eingeführt werden, um die Verarbeitungszeit und Reaktionsgeschwindigkeit auf Arbitragemöglichkeiten zu reduzieren.

### **7.2.2 Einsatz von Websockets**

Statt auf [REST APIs](#) der Börsen zuzugreifen, könnte die Anwendung Websockets nutzen, um die Orderbücher und die Geld- und Briefkurse schneller zu erhalten. Durch die schnellere Datenübertragung könnten Arbitragemöglichkeiten effizienter erkannt und ausgenutzt werden.

### **7.2.3 Integration weiterer Börsen**

Eine Erweiterung der Anwendung um zusätzliche Kryptowährungsbörsen, insbesondere internationale Börsen, könnte zu mehr Arbitragemöglichkeiten und höheren potenziellen [RoTs](#) führen. Bessere Gebührenstrukturen bei diesen Börsen könnten

die Rentabilität von Arbitragegeschäften erhöhen. Allerdings sollte beachtet werden, dass die Integration zusätzlicher Börsen auch Risiken mit sich bringt. Regulierungen und Börsenschließungen, wie im Fall von FTX, oder der Rückzug von Börsen aus bestimmten Märkten, wie Crypto.com aus Deutschland [84], können das Risiko bei der Nutzung solcher Börsen erhöhen. Daher ist es wichtig, die regulatorischen Rahmenbedingungen und die Stabilität der ausgewählten Börsen zu berücksichtigen.

## 7.3 Zusammenfassung

Die Bachelorthesis „Anforderungsanalyse und Design einer Arbitrage-Trading Zielarchitektur für Kryptowährungen“ hat zum Großteil eine Anwendung entwickelt, die Arbitragemöglichkeiten in Echtzeit erkennen und protokollieren kann. Durch den Vergleich der Geld- und Briefkurse von drei der größten Kryptowährungsbörsen, Binance, Bitstamp und Coinbase Pro, hat die Anwendung das Potenzial, dem Benutzer dabei zu helfen, von Marktineffizienzen zu profitieren.

Allerdings zeigt die Analyse der Testergebnisse, dass sich Arbitragemöglichkeiten in unserem Fall nicht rentieren. Die Durchlaufzeit je Trade von ca. 0,8 - 1,2 Sekunden ist als zu lang zu werten, da die Arbitragemöglichkeiten in den meisten Fällen schnell wieder geschlossen werden. Des Weiteren macht es die Gebührenstruktur der Börsen, insbesondere Coinbase mit 0,6 % Gebühren für Taker, schwierig, diese für Arbitragegeschäfte zu nutzen. Die Preise der Kurse müssen erheblich voneinander abweichen, damit eine Börse mit prozentual hohen Gebühren infrage kommt. Aus den Protokolldateien geht hervor, dass Coinbase zu keinem Zeitpunkt von der Handelslogik als geeignete Börse für Arbitragegeschäfte in Betracht gezogen wurde. Selbst wenn Arbitragemöglichkeiten entstehen, ist der **RoT** pro Handel nicht hoch und das Risiko, den Handel vollständig durchzuführen, ist hingegen nicht tragbar. Um die tatsächliche Rentabilität von Arbitragegeschäften zu bestimmen, müssten realitätsnahe Tests durchgeführt werden, die zeigen, dass diese Arbitragemöglichkeiten tatsächlich ausgenutzt werden können. Angesichts der oben genannten Durchlaufzeit ist dies jedoch sehr unwahrscheinlich.

Testweise wurde die Anwendung im Zeitraum vom 14.03.2023 bis 20.03.2023 mit unterschiedlichen Laufzeiten ausgeführt. Bei diesem Testlauf lief die Anwendung mit Gebühren von 0,1 % für Binance und 0 % für Bitstamp, da bis zu einem Handelsvolumen von unter 1.000 US-Dollar keine Gebühren anfallen [85]. Insgesamt wurden 700 Arbitragemöglichkeiten zwischen Binance und Bitstamp identifiziert und ein möglicher Gesamtprofit von ca. 1204,73 **USDT** aufgezeichnet. Der kumulierte Einsatz bei dem möglichen Gesamtprofit wäre ca. 2.296.256,77 **USDT** gewesen. Zu beachten

ist die nicht vorhandene Gebühr bei Bitstamp, welche den Gesamtprofit deutlich verfälscht. Allerdings kann für den Anwendungsfall, dass ein Trade gebührenfrei vollzogen werden kann, beobachtet werden, dass drei Arbitragemöglichkeiten mit größeren **RoTs**, nämlich 192,65 **USDT**, 193,17 **USDT** und 194,45 **USDT** protokolliert wurden. Es ist wichtig zu berücksichtigen, dass der Ertrag dieser Trades nicht durch Preisschwankungen, sondern durch das Handelsvolumen zustande kam. Bei diesen Trades lag der Einsatz je Trade bei über 370.000 **USDT**.

Ein weiterer Testlauf mit angepasster Taker Gebühr von 0,3 % bei Bitstamp [85] wurde vom 31.03.2023 um 03:35:11 gestartet und am 01.04.2023 um 20:46:52 Uhr beendet. Hierbei wurden keine Arbitragemöglichkeiten identifiziert. Dieses Testergebnis zeigt die Herausforderung mit hohen Gebührenstrukturen auf, sodass Arbitragemöglichkeiten erst wahrgenommen werden können, wenn die Preise auf den zu vergleichenden Börsen erheblich voneinander abweichen müssen.

Insgesamt zeigt diese wissenschaftliche Ausarbeitung, dass die entwickelte Anwendung zwar grundsätzlich funktioniert, aber aufgrund von Latenz und Gebührenstrukturen in der Praxis Rentabilitätssteigerungen bedarf. Verbesserungen in der Ausführungszeit, die Implementierung automatisierter Handelsfunktionen sowie die Integration von Börsen mit niedrigeren Gebühren könnten jedoch dazu beitragen, die Anwendung in der Zukunft effektiver zu gestalten. Trotz der festgestellten Einschränkungen bietet die Arbeit eine solide Grundlage für zukünftige Weiterentwicklungen im Bereich der Arbitrage-Trading Zielarchitektur für Kryptowährungen.

# Kapitel 8

## Anhang

Die entwickelte Anwendung sowie alle .log-Dateien sind unter [\[73\]](#) verfügbar.

Einlesen von .csv-Dateien für die Tabelle Binance:

```
COPY binanceEUR(unwichtig, price, amount, quote_amount, unix_time,
unwichtigbool, unwichtigbool_zwei)
FROM '/private/tmp/binanceEUR.csv'
DELIMITER ','
CSV HEADER;

BEGIN;

ALTER TABLE binanceUSD RENAME TO binanceUSD_old;

CREATE TABLE binanceUSD (
    id SERIAL,
    price NUMERIC,
    amount NUMERIC,
    unix_time BIGINT,
    exchange VARCHAR(99) DEFAULT 'binanceUSD' NOT NULL,
    PRIMARY KEY (id, exchange)
);

INSERT INTO binanceUSD (id, price, amount, unix_time, exchange)
SELECT id, price, amount, unix_time, exchange FROM binanceUSD_old;

DROP TABLE binanceUSD_old;
COMMIT;
```

Instruktionen für die Tabelle Binance:

```
/* Umbenennen der Tabelle, damit sie nicht verloren geht */
ALTER TABLE binanceUSD RENAME TO binanceUSD_old;

/* Erzeugen der neuen Tabelle mit benötigten Attributen
aus der Ursprungstabelle */
CREATE TABLE binanceUSD (
    id SERIAL,
    price NUMERIC,
    amount NUMERIC,
    unix_time BIGINT,
    exchange VARCHAR(99) DEFAULT 'binanceUSD' NOT NULL,
    PRIMARY KEY (id, exchange)
);

/* Einfuegen der gesicherten Werte in die neue Tabelle */
INSERT INTO binanceUSD (id, price, amount, unix_time, exchange)
SELECT id, price, amount, unix_time, exchange FROM binanceUSD_old;

/* Entfernen der alten Tabelle */
DROP TABLE binanceUSD_old;
```

Hinzufügen der benötigten Attribute **price\_after\_fee** und **price\_and\_amount**

```
/* Hinzufügen der beiden Spalten für binanceUSD*/
ALTER TABLE binanceUSD
ADD COLUMN price_after_fee NUMERIC,
ADD COLUMN price_and_amount NUMERIC;

/* Einsetzen der Werte für die Spalten */
UPDATE binanceUSD
SET price_after_fee = price * 1.001,
    price_and_amount = amount * (price * 1.001);

/* Hinzufügen der beiden Spalten für coinbaseUSD*/
ALTER TABLE coinbaseUSD
ADD COLUMN price_after_fee NUMERIC,
ADD COLUMN price_and_amount NUMERIC;

/* Einsetzen der Werte für die Spalten */
```

```

UPDATE coinbaseUSD
SET price_after_fee = price * 1.006,
    price_and_amount = amount * (price * 1.006);

/* Hinzufügen der beiden Spalten für bitstampUSD*/
ALTER TABLE bitstampUSD
ADD COLUMN price_after_fee NUMERIC,
ADD COLUMN price_and_amount NUMERIC;

/* Einsetzen der Werte für die Spalten */
UPDATE bitstampUSD
SET price_after_fee = price,
    price_and_amount = amount * price;

```

Zusammenführen der drei Tabellen Binance, Bitstamp und Coinbase:

```

SELECT
    c.unix_time AS coinbase_unix_time,
    c.price AS coinbase_price,
    c.price_after_fee AS coinbase_price_after_fee,
    c.amount AS coinbase_amount,
    c.price_and_amount AS coinbase_price_and_amount,
    b.unix_time AS bitstamp_unix_time,
    b.price AS bitstamp_price,
    b.price_after_fee AS bitstamp_price_after_fee,
    b.amount AS bitstamp_amount,
    b.price_and_amount AS bitstamp_price_and_amount,
    bn.unix_time AS binance_unix_time,
    bn.price AS binance_price,
    bn.price_after_fee AS binance_price_after_fee,
    bn.amount AS binance_amount,
    bn.price_and_amount AS binance_price_and_amount
FROM
    coinbaseUSD c
FULL OUTER JOIN
    bitstampUSD b
ON
    c.unix_time = b.unix_time
FULL OUTER JOIN
    binanceUSD bn
ON

```

```
        c.unix_time = bn.unix_time
WHERE
        c.unix_time = b.unix_time
    OR c.unix_time = bn.unix_time
    OR b.unix_time = bn.unix_time
ORDER BY
        COALESCE(c.unix_time, b.unix_time, bn.unix_time);
```



# Literaturverzeichnis

- [1] Herbert Sperber. *Finanzmärkte: eine praxisorientierte Einführung*. ISBN: 9783791033952 Place: Stuttgart. 2014. URL: [http://deposit.d-nb.de/cgi-bin/dokserv?id=4671882&prov=M&dok\\_var=1&dok\\_ext=htm](http://deposit.d-nb.de/cgi-bin/dokserv?id=4671882&prov=M&dok_var=1&dok_ext=htm).
- [2] Martin Schwarz. *Neue Kryptowährungen 2023 Neue Coins mit Potenzial kaufen*. URL: <https://coincierge.de/kryptowaehrungen-kaufen/neue-kryptowaehrungen/> (besucht am 01.04.2023).
- [3] Arvind Narayanan u. a. *Bitcoin and cryptocurrency technologies : a comprehensive introduction*. eng. 2016. ISBN: 978-0-691-17169-2. URL: [http://bvbr.bib-bvb.de:8991/F?func=service&doc\\_library=BVB01&local\\_base=BVB01&doc\\_number=029117052&line\\_number=0001&func\\_code=DB\\_RECORDS&service\\_type=MEDIA](http://bvbr.bib-bvb.de:8991/F?func=service&doc_library=BVB01&local_base=BVB01&doc_number=029117052&line_number=0001&func_code=DB_RECORDS&service_type=MEDIA).
- [4] Huawei Huang u. a. „A Survey of State-of-the-Art on Blockchains“. In: *ACM Computing Surveys* 54.2 (2022). Publisher: Association for Computing Machinery (ACM), S. 1–42. ISSN: 0360-0300. DOI: [10.1145/3441692](https://doi.org/10.1145/3441692). URL: <https://dx.doi.org/10.1145/3441692>.
- [5] Satoshi Nakamoto. „Bitcoin: A Peer-to-Peer Electronic Cash System“. en. In: (2008).
- [6] Meltem Demirors, Cuy Sheffield und World Economic Forum’s Global Future Council on Cryptocurrencies. „Crypto, What Is It Good For? An Overview of Cryptocurrency Use Cases“. In: *World Economic Forum* (Dez. 2020), S. 2–37. URL: [https://www3.weforum.org/docs/WEF\\_Cryptocurrency\\_Uses\\_Cases\\_2020.pdf](https://www3.weforum.org/docs/WEF_Cryptocurrency_Uses_Cases_2020.pdf).
- [7] Florian Tschorsch und Bjorn Scheuermann. „Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies“. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016). Publisher: Institute of Electrical and Electronics Engineers (IEEE), S. 2084–2123. ISSN: 1553-877X. DOI: [10.1109/comst.2016.2535718](https://doi.org/10.1109/comst.2016.2535718). URL: <https://dx.doi.org/10.1109/comst.2016.2535718>.
- [8] Marco Iansiti und Karim R. Lakhani. „The Truth About Blockchain“. In: *Harvard Business Review* (Jan. 2017). Section: Blockchain. ISSN: 0017-8012. URL: <https://hbr.org/2017/01/the-truth-about-blockchain> (besucht am 19.03.2023).
- [9] Paul Müller u. a. „The Bitcoin Universe: An Architectural Overview of the Bitcoin Blockchain“. In: *11. DFN-Forum Kommunikationstechnologien*. Hrsg. von Paul Müller u. a. Bonn: Gesellschaft für Informatik e.V., 2018, S. 1–20.
- [10] Michael Doran. „A forensic look at Bitcoin cryptocurrency“. In: 2014.
- [11] Chris Burniske. „ARK INVEST IN COLLABORATION WITH DR. ARTHUR B. LAFFER AND LAFFER ASSOCIATES“. en. In: *ARK Invest Research* (2015).

- [12] Mauro Conti u. a. „A Survey on Security and Privacy Issues of Bitcoin“. In: *IEEE Communications Surveys & Tutorials* 20.4 (2018). Publisher: Institute of Electrical and Electronics Engineers (IEEE), S. 3416–3452. ISSN: 1553-877X. DOI: [10.1109/comst.2018.2842460](https://doi.org/10.1109/comst.2018.2842460). URL: <https://dx.doi.org/10.1109/comst.2018.2842460>.
- [13] Wolfgang K. Härdle, Campbell R. Harvey und Raphael C. G. Reule. „Understanding Cryptocurrencies“. In: *SSRN Electronic Journal* (2019). Publisher: Elsevier BV. ISSN: 1556-5068. DOI: [10.2139/ssrn.3360304](https://doi.org/10.2139/ssrn.3360304). URL: <https://dx.doi.org/10.2139/ssrn.3360304>.
- [14] Scott Driscoll. *How Bitcoin Works Under the Hood*. Juli 2013. URL: <http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html> (besucht am 23.03.2023).
- [15] *Blockchain - IPD - Institut für Prozessmanagement und Digitale Transformation - FH Münster*. Apr. 2022. URL: <https://www.fh-muenster.de/ipd/a-z/blockchain.php> (besucht am 23.03.2023).
- [16] Saskia Littmann. „El Salvador: Lieber Bitcoin als Dollar“. de. In: (Sep. 2021). URL: <https://www.wiwo.de/finanzen/boerse/el-salvador-lieber-bitcoin-als-dollar-/27585876.html> (besucht am 23.03.2023).
- [17] Andreas Hanl und Jochen Michaelis. „Kryptowährungen — ein Problem für die Geldpolitik?“ de. In: *Wirtschaftsdienst* 97.5 (Mai 2017), S. 363–370. ISSN: 0043-6275, 1613-978X. DOI: [10.1007/s10273-017-2145-y](https://doi.org/10.1007/s10273-017-2145-y). URL: <http://link.springer.com/10.1007/s10273-017-2145-y> (besucht am 23.03.2023).
- [18] Vitalik Buterin. „Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.“ en. In: (2014).
- [19] Philipp Schuster, Erik Theissen und Marliese Uhrig-Homburg. „Finanzwirtschaftliche Anwendungen der Blockchain-Technologie“. In: *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung* 72.2 (2020). Publisher: Springer Science and Business Media LLC, S. 125–147. ISSN: 0341-2687. DOI: [10.1007/s41471-020-00090-5](https://doi.org/10.1007/s41471-020-00090-5). URL: <https://dx.doi.org/10.1007/s41471-020-00090-5>.
- [20] Sherman Lee. „Bitcoin’s Energy Consumption Can Power An Entire Country – But EOS Is Trying To Fix That“. en. In: *Forbes* (Apr. 2018). Section: Asia. URL: <https://www.forbes.com/sites/shermanlee/2018/04/19/bitcoins-energy-consumption-can-power-an-entire-country-but-eos-is-trying-to-fix-that/> (besucht am 19.03.2023).
- [21] Jeremy Hinsdale. *Cryptocurrency’s Dirty Secret: Energy Consumption*. en. Mai 2022. URL: <https://news.climate.columbia.edu/2022/05/04/cryptocurrency-energy/> (besucht am 19.03.2023).
- [22] tagesschau.de. *Finanzmarktexperte zu Bitcoin & Co.: "Kryptobörsen sind unsicher"*. de. Jan. 2023. URL: <https://www.tagesschau.de/wirtschaft/finanzen/kryptoboersen-regulierung-crashes-anleger-investition-101.html> (besucht am 26.03.2023).
- [23] Dirk G. Baur und Thomas Dimpfl. „The volatility of Bitcoin and its role as a medium of exchange and a store of value“. In: *Empirical Economics* 61.5 (2021). Publisher: Springer Science and Business Media LLC, S. 2663–2683. ISSN: 0377-7332. DOI: [10.1007/s00181-020-01990-5](https://doi.org/10.1007/s00181-020-01990-5). URL: <https://dx.doi.org/10.1007/s00181-020-01990-5>.

- [24] Axel Kannenberg. *Studie: Geldwäsche mit Kryptowährungen 2021 um 30 Prozent gestiegen*. de. Jan. 2022. URL: <https://www.heise.de/news/Studie-Geldwaesche-mit-Kryptowaehrungen-2021-um-30-Prozent-gestiegen-6341024.html> (besucht am 26.03.2023).
- [25] *Regulierung von Kryptowährung im Überblick - IDnow*. de-DE. URL: <https://www.idnow.io/de/regulatorik/krypto-regulierung/> (besucht am 26.03.2023).
- [26] David Scheider. *Heute vor 13 Jahren: Erste Bitcoin-Transaktion*. de-DE. Jan. 2022. URL: <https://www.btc-echo.de/news/heute-vor-13-jahren-erste-bitcoin-transaktion-132758/> (besucht am 20.03.2023).
- [27] *Cryptocurrency Prices, Charts And Market Capitalizations*. en. URL: <https://coinmarketcap.com/> (besucht am 30.03.2023).
- [28] *Bitcoin-Halving*. de. URL: <https://www.ig.com/de/bitcoin-handel/bitcoin-halving> (besucht am 30.03.2023).
- [29] Sören Imöhl. *Bitcoin-Kurs (BTC) aktuell: Bitcoin markiert neues Jahreshoch – Kurs steigt über 28.000-Dollar-*. de. März 2023. URL: <https://www.wiwo.de/finanzen/boerse/bitcoin-kurs-btc-aktuell-bitcoin-markiert-neues-jahreshoch-kurs-steigt-ueber-28-000-dollar-marke/27382428.html> (besucht am 20.03.2023).
- [30] *Can Bitcoin's Hard Cap of 21 Million Be Changed? | River Learn - Bitcoin Basics*. en-us. Section: River Learn - Bitcoin Basics. URL: <https://river.com/learn/can-bitcoins-hard-cap-of-21-million-be-changed/> (besucht am 20.03.2023).
- [31] *Tether price today, USDT to USD live, marketcap and chart*. en. URL: <https://coinmarketcap.com/currencies/tether/> (besucht am 18.03.2023).
- [32] *Tether Transparency*. URL: <https://tether.to/en/transparency/> (besucht am 20.03.2023).
- [33] Gordon Y. Liao und John Caramichael. „Stablecoins: Growth Potential and Impact on Banking“. en. In: *International Finance Discussion Paper 2022.1334* (Jan. 2022), S. 1–26. ISSN: 10732500, 27674509. DOI: [10.17016/IFDP.2022.1334](https://doi.org/10.17016/IFDP.2022.1334). URL: <https://www.federalreserve.gov/econres/ifdp/stablecoins-growth-potential-and-impact-on-banking.htm> (besucht am 20.03.2023).
- [34] Jitender Tokas. *Digital Derivatives – The Rise of USDT and DeFi*. en. 2020. URL: <https://www.delta.exchange/blog/digital-derivatives-the-rise-of-usdt-and-defi> (besucht am 20.03.2023).
- [35] Douglas W. Arner, Raphael Auer und Jon Frost. „Stablecoins: Risks, Potential and Regulation“. In: *SSRN Electronic Journal* (2020). Publisher: Elsevier BV. ISSN: 1556-5068. DOI: [10.2139/ssrn.3979495](https://doi.org/10.2139/ssrn.3979495). URL: <https://dx.doi.org/10.2139/ssrn.3979495>.
- [36] *why use tether instead of usd | BTCC Knowledge*. en-US. URL: <https://www.btcc.com/en-US/hashtag/why-use-tether-instead-of-usd> (besucht am 20.03.2023).
- [37] Garrick Hileman und Michel Rauchs. „2017 Global Cryptocurrency Benchmarking Study“. In: *SSRN Electronic Journal* (2017). Publisher: Elsevier BV, S. 22. ISSN: 1556-5068. DOI: [10.2139/ssrn.2965436](https://doi.org/10.2139/ssrn.2965436). URL: <https://dx.doi.org/10.2139/ssrn.2965436>.

- [38] Oliver Meissner-Wollnitz. *Kryptobörsen im Vergleich 2021 Die besten Anbieter im Test*. de-DE. März 2023. URL: <https://www.handelsblatt.com/vergleich/krypto-boersen-vergleich/> (besucht am 20.03.2023).
- [39] Jens Kerkmann. *CEX vs DEX – hier liegen die Unterschiede*. de-DE. Dez. 2020. URL: <https://blockchainwelt.de/cex-vs-dex-hier-liegen-die-unterschiede/> (besucht am 20.03.2023).
- [40] Tyler Moore und Nicolas Christin. „Beware the Middleman: Empirical Analysis of Bitcoin-Exchange Risk“. In: *Financial Cryptography and Data Security*. Hrsg. von Ahmad-Reza Sadeghi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 25–33. ISBN: 978-3-642-39884-1. DOI: [10.1007/978-3-642-39884-1\\_3](https://dx.doi.org/10.1007/978-3-642-39884-1_3). URL: [https://dx.doi.org/10.1007/978-3-642-39884-1\\_3](https://dx.doi.org/10.1007/978-3-642-39884-1_3).
- [41] Mirko Dölle. *Unternehmenskontrolle versagt*. de. Dez. 2022. URL: <https://www.heise.de/news/Die-Milliarden-Pleite-der-Kryptowährungsboerse-FTX-und-ihre-Folgen-7359608.html> (besucht am 28.03.2023).
- [42] Philipp Henk. *FTX: Insolvenzverfahren, Auswirkungen, Updates (2023)*. de. Feb. 2023. URL: <https://bitcoin-2go.de/ftx-insolvenz/> (besucht am 28.03.2023).
- [43] James Howell. *Top 10 DeFi Hacks You Should Know*. en-US. Dez. 2022. URL: <https://101blockchains.com/top-defi-hacks/> (besucht am 28.03.2023).
- [44] *Comprehensive List of DeFi Hacks & Exploits*. en-US. März 2023. URL: <https://chainsec.io/defi-hacks/> (besucht am 28.03.2023).
- [45] Dr Philipp Giese. *Was ist das Order Book? Das Werkzeug der Bitcoin-Trader*. de-DE. Dez. 2018. URL: <https://www.btc-echo.de/news/was-ist-das-order-book-19357/> (besucht am 02.04.2023).
- [46] Lukas Mantinger. *Für Einsteiger: So funktionieren Börsen- Kryptowährungen kaufen*. de-DE. Juni 2019. URL: <https://cryptoticker.io/de/kryptoboersen-funktionieren/> (besucht am 02.04.2023).
- [47] Bitpanda. *Maker-Gebühren und Taker-Gebühren beim Kryptotrading*. de-DE. URL: <https://www.bitpanda.com/academy/de/lektionen/was-sind-maker-gebuehren-und-taker-gebuehren-beim-traden-mit-kryptowahrungen> (besucht am 02.04.2023).
- [48] UnicornFinance. *Orderbuch erklärt*. de-DE. Feb. 2022. URL: <https://www.unicornfinance.de/orderbuch-erklaert/> (besucht am 02.04.2023).
- [49] *Arbitrage: Was ist das? (Definition & Infos)*. de. Okt. 2016. URL: <https://www.financescout24.de/wissen/ratgeber/arbitrage> (besucht am 20.02.2023).
- [50] Katrin Alisch und Thorsten Hader, Hrsg. *Gabler-Wirtschafts-Lexikon [die ganze Welt der Wirtschaft ; Betriebswirtschaft, Volkswirtschaft, Recht, Steuern] : A - D*. Edition: 15., vollständig überarbeitete und aktualisierte Auflage Place: Wiesbaden. 2000. URL: [http://scans.hebis.de/HEBCGI/show.pl?09396527\\_toc.pdf](http://scans.hebis.de/HEBCGI/show.pl?09396527_toc.pdf).
- [51] Bundeszentrale für politische Bildung. *Arbitrage*. de. URL: <https://www.bpb.de/kurz-knapp/lexika/lexikon-der-wirtschaft/18703/arbitrage/> (besucht am 18.02.2023).

- [52] Bert Rürup. *Fischer-Wirtschaftslexikon*. Edition: Orig.-Ausg., 3., aktualisierte und erw. Aufl. ISBN: 3596140811 Place: Frankfurt am Main Series: Fischer-Taschenbücher Volume: 14081 : Wirtschaft. 1998. URL: [http://scans.hebis.de/HEBCGI/show.pl?05970607\\_toc.pdf](http://scans.hebis.de/HEBCGI/show.pl?05970607_toc.pdf).
- [53] Erwin Dichtl, Hrsg. *Vah lens großes Wirtschaftslexikon : 1. A - K*. Edition: 2., überarb. und erw. Aufl. Place: München. 1993. URL: [http://scans.hebis.de/HEBCGI/show.pl?02556779\\_toc.pdf](http://scans.hebis.de/HEBCGI/show.pl?02556779_toc.pdf).
- [54] Silvia Zhang. *Krypto-Crash 2022: Der Zusammenhang zwischen Zinserhöhungen und Krypto-Kursen*. de-DE. Okt. 2022. URL: <https://phemex.com/de/academy/krypto-crash-2022-der-zusammenhang-zwischen-zinserhoehungen-und-krypto-kursen> (besucht am 02.04.2023).
- [55] *Was ist Krypto Arbitrage?* de. URL: <https://capital.com/de/was-ist-krypto-arbitrage> (besucht am 01.04.2023).
- [56] Portara. *Tick Data – What It Is and Why It’s Essential for Traders*. en-US. Feb. 2023. URL: <https://portaracqg.com/2023/02/01/what-is-tick-data/> (besucht am 29.03.2023).
- [57] *Buy/Sell Bitcoin, Ether and Altcoins | Cryptocurrency Exchange*. URL: <https://www.binance.com> (besucht am 29.03.2023).
- [58] Krisztian Buza, Gábor I. Nagy und Alexandros Nanopoulos. „Storage-optimizing clustering algorithms for high-dimensional tick data“. In: *Expert Systems with Applications* 41.9 (2014). Publisher: Elsevier BV, S. 4148–4157. ISSN: 0957-4174. DOI: [10.1016/j.eswa.2013.12.046](https://doi.org/10.1016/j.eswa.2013.12.046). URL: <https://dx.doi.org/10.1016/j.eswa.2013.12.046>.
- [59] *The Importance of Good Data Sets When Backtesting (Garbage In Equals Garbage Out) – Quantified Strategies For Traders And Investors*. en-US. Feb. 2023. URL: <https://www.quantifiedstrategies.com/the-importance-of-good-data-sets/> (besucht am 01.04.2023).
- [60] *The most granular data for cryptocurrency markets*. en. URL: <https://tardis.dev> (besucht am 01.04.2023).
- [61] Janelle Veasey. *Transforming backtesting data*. en. Section: Future of Investing & Trading. Sep. 2020. URL: <https://www.refinitiv.com/perspectives/future-of-investing-trading/transforming-backtesting-data/> (besucht am 01.04.2023).
- [62] Vishesh Aurora. *How to reduce memory usage in Python (Pandas)?* Apr. 2021. URL: <https://www.analyticsvidhya.com/blog/2021/04/how-to-reduce-memory-usage-in-python-pandas/> (besucht am 01.04.2023).
- [63] Phil Allsopp. *Where and how does PostgreSQL actually store data on disk?* en-us. URL: <https://www.postgresql.fastware.com/blog/where-and-how-does-postgresql-actually-store-data-on-disk> (besucht am 01.04.2023).
- [64] *8.14. JSON Types*. en. Feb. 2023. URL: <https://www.postgresql.org/docs/15/datatype-json.html> (besucht am 01.04.2023).
- [65] Bernhard Knasmüller. *Storing Semi-Structured Data in PostgreSQL*. en-US. Juli 2020. URL: <https://knasmueller.net/storing-semi-structured-data-in-postgresql> (besucht am 01.04.2023).



- [66] Team Devathon. *SQL vs NoSQL comparison: MySQL, PostgreSQL, MongoDB & Cassandra*. en-US. März 2021. URL: <https://devathon.com/blog/sql-vs-nosql-mysql-vs-postgresql-vs-mongodb-vs-cassandra/> (besucht am 01.04.2023).
- [67] Igor Makarov und Antoinette Schoar. „Trading and arbitrage in cryptocurrency markets“. In: *Journal of Financial Economics* 135.2 (2020). Publisher: Elsevier BV, S. 293–319. ISSN: 0304-405X. DOI: [10.1016/j.jfineco.2019.07.001](https://doi.org/10.1016/j.jfineco.2019.07.001). URL: <https://dx.doi.org/10.1016/j.jfineco.2019.07.001>.
- [68] *Pricing | CryptoCompare Cryptocurrency Data API*. en. URL: <https://min-api.cryptocompare.com/pricing> (besucht am 02.04.2023).
- [69] *Automatic Trading - Features*. en-us. URL: <https://www.cryptohopper.com/> (besucht am 02.04.2023).
- [70] *How to Create a Crypto Arbitrage Trading Bot?* en. URL: <https://maticz.com/how-to-build-a-crypto-arbitrage-bot> (besucht am 02.04.2023).
- [71] Paco. *AMM Arbitrageur*. original-date: 2021-04-08T09:48:22Z. Apr. 2023. URL: <https://github.com/paco0x/amm-arbitrageur> (besucht am 02.04.2023).
- [72] *Bitcoincharts | Markets API*. URL: <https://bitcoincharts.com/about/markets-api/> (besucht am 26.03.2023).
- [73] Bugday. *GitHub Repository zur Thesis*. URL: <https://github.com/ebugday81/arbitrabort>. (letzter Zugriff: 31.03.2023).
- [74] *binance/binance-public-data*. en. URL: <https://github.com/binance/binance-public-data> (besucht am 27.03.2023).
- [75] *pgAdmin - PostgreSQL Tools*. URL: <https://www.pgadmin.org/> (besucht am 27.03.2023).
- [76] Antonio. *Answer to "Postgres ERROR: could not open file for reading: Permission denied"*. Sep. 2017. URL: <https://stackoverflow.com/a/46477635/10224308> (besucht am 27.03.2023).
- [77] user637338. *Answer to "Postgres ERROR: could not open file for reading: Permission denied"*. Apr. 2015. URL: <https://stackoverflow.com/a/29899948/10224308> (besucht am 27.03.2023).
- [78] Daniel Paquin. *cbpro: The unofficial Python client for the Coinbase Pro API*. URL: <https://github.com/danpaquin/coinbasepro-python> (besucht am 31.03.2023).
- [79] Igor Kroitor. *ccxt: A JavaScript / Python / PHP cryptocurrency trading library with support for 130+ exchanges*. URL: <https://ccxt.com> (besucht am 31.03.2023).
- [80] Kamil Madac. *BitstampClient: Bitstamp API python implementation*. URL: <https://github.com/kmadac/bitstamp-python-client> (besucht am 31.03.2023).
- [81] *General API Information – Binance API Documentation*. URL: <https://binance-docs.github.io/apidocs/spot/en/#general-api-information> (besucht am 31.03.2023).
- [82] *Bitstamp Trusted Crypto Exchange | Buy & Sell Cryptocurrencies*. en. URL: <https://www.bitstamp.net/api/> (besucht am 31.03.2023).

- [83] *Download von PyCharm: Python-IDE von JetBrains für professionelle Entwickler.* de. URL: <https://www.jetbrains.com/de-de/pycharm/download/> (besucht am 31.03.2023).
- [84] John Stanley Hunter. *Crypto.com zieht sich aus Deutschland zurück – Bafin ermittelt.* de. März 2023. URL: <https://financefwd.com/de/crypto-com-deutschland-rueckzug/> (besucht am 02.04.2023).
- [85] *Fee schedule.* en. URL: <https://www.bitstamp.net/fee-schedule/> (besucht am 01.04.2023).