**Crypto-Market Engine**

UML Projekt

Object Oriented Analysis

BWI A-16

Autoren

Michael Gantenbein, Nicolai Brunner

Michèle Troller-Wick, Simona Patrut

6. Februar 2018

Dozent

Stefan Berger

**Inhaltsverzeichnis**

[*1* *Auftrag* 2](#_Toc505614020)

[*2* *Projektbeschreibung* 3](#_Toc505614021)

[2.1 Auswahl Applikations-Lösung 3](#_Toc505614022)

[2.2 Methodik für Umsetzung 3](#_Toc505614023)

[2.3 Vorbereitung und unterstützende Medien 3](#_Toc505614024)

[*3* *Projektablauf* 4](#_Toc505614025)

[*4* *Spezifikation der Anforderungen* 5](#_Toc505614026)

[4.1 Verhaltens-Diagramme / behaviour diagrams 6](#_Toc505614027)

[4.1.1 Use Case Diagramm 6](#_Toc505614028)

[4.1.2 Beschreibung Use Cases und Aktivitätsdiagramme 6](#_Toc505614029)

[4.1.2.1 URL aufrufen 7](#_Toc505614030)

[4.1.2.3 Anlage platzieren 8](#_Toc505614031)

[4.1.2.4 GS Tracker / Kill-Switch 9](#_Toc505614032)

[4.1.2.5 Engine Arbitrage: Algorithmus 1 – Fill The Gap 9](#_Toc505614033)

[4.1.2.6 Engine PaiRiT: manuell traden – Pair the Pair 10](#_Toc505614034)

[4.1.2.7 Tracker auszahlen 11](#_Toc505614035)

[4.1.2.8 Engine Delta Pips: Algorithmus #3 / Bonus 11](#_Toc505614036)

[4.2 Struktur-Diagramm / structure diagram 12](#_Toc505614037)

[4.2.1 Klassen Diagramm 12](#_Toc505614038)

[4.2.2 Verteilungsdiagramm / Deployment Diagram 13](#_Toc505614039)

[*5* *Technische Umsetzung* 13](#_Toc505614040)

[5.1 Design / Entwicklung 13](#_Toc505614041)

[5.2 Datenbank / Plattform 13](#_Toc505614042)

[5.3 Testen / Verbessern / Implementieren 14](#_Toc505614043)

[*6* *Zusammenfassung / Schlusswort* 14](#_Toc505614044)

[*7* *Glossar* 15](#_Toc505614045)

[*8* *Quellenverzeichnis* 19](#_Toc505614046)

# *Auftrag*

Der Auftrag für die «Case-Arbeit» des Kurses Object Oriented Analysis der Studiengruppe BWI-A16 besteht darin, eine softwarebasierte Lösung mittels UML (Unified Modeling Language) Technik zu erarbeiten, zu dokumentieren, mit dem Programm Java zu realisieren sowie im Anschluss das Programm auf einer unabhängigen Plattform zu demonstrieren.

Das während der Kurse angeeignete Wissen über die Objekt-Orientierte Denkweise (Klassen, Assoziationen und Strukturen) sowie über die qualitativen Anforderungen an «Requirements» und deren Dokumentation (Use Cases, Aktivitätsdiagramme und Klassendiagramm) sind für die Umsetzung des Auftrages essentiell.

# *Projektbeschreibung*

## Auswahl Applikations-Lösung

Die im Semesterprogramm von Herrn Berger als mögliche Themen vorgeschlagenen Beispiele wurden kurz diskutiert. Aufgrund der vertieften Kenntnisse und Interessen Herrn Gantenbeins zum Thema «Handel mit Kryptowährungen» sowie einer bereits sehr konkreten Idee seinerseits, wurde die Auswahl für das umzusetzende Programm schnell und demokratisch (3 zu 2) gefällt.

## Methodik für Umsetzung

Das beschriebene Projekt wird in zwei Teile unterteilt. Einerseits wird die UML Methoden verwendet um Diagramme aufzuzeichnen sowie um einen Überblick zu verschaffen, über welche Bedingungen und Eigenschaften das Programm verfügen soll. Andererseits wird im zweiten Teil der Quellcode mit den nötigen Schnittstellen programmiert und getestet.

Der UML Teil umfasst die komplette theoretische Umsetzung der Applikation. Dieser beinhaltet Use Cases, Aktivitätsdiagramme und ein Klassendiagramm, sowie Beschreibungen der einzelnen Funktionen der Applikation.

Zu einem späteren Zeitpunkt werden der Quellcode und die technischen Erläuterungen zum Programm erstellt und präsentiert. Durch die technische Realisation der im Vorhinein erarbeiteten Grundlagen werden Aktivitätsdiagramme und Klassendiagramm auf Realisierbarkeit und Umsetzung geprüft.

## Vorbereitung und unterstützende Medien

Aufgrund der noch fehlenden Programmier- sowie Modellier-Erfahrung der Gruppe, konnten die vorhandene Kompetenzen der Gruppenmitglieder noch nicht konkret definiert und dem Projekt zugeordnet werden. Dies stellte sich nachträglich als zusätzliche Herausforderung heraus.

Als unterstützende Programme wurden schnell «GitHub» und «Lucidchart» gewählt. Zudem wurde für die End-Dokumentation das Programm «Overleaf» vorgeschlagen. Die fehlende Erfahrung der Gruppenmitglieder mit diesen Programmen stellte sich jedoch während des Projektverlaufs als etwas nachteilig heraus. So wurde zum Beispiel erst später erkannt, dass das Programm Visio optimalere Eigenschaften für die UML Dokumentation aufweist. Auch die Dokumentation wurde aufgrund fehlender zeitlicher Ressourcen schlussendlich im Word erstellt. Die restliche Kommunikation zwischen den Projektteilnehmern verlief mittels «WhatsApp», Email und «Dropbox».

# *Projektablauf*

Um den Auftrag koordiniert durchzuführen zu können, wurde der Projektablauf vorab definiert. Wie bereits erwähnt waren die Kompetenzen der Projektteilnehmer, unter anderem hinsichtlich Programmierung, zu diesem Zeitpunkt noch nicht bekannt. Die einzelnen Arbeitsschritte konnten demnach erst im Verlauf des Projektes den Mitgliedern zugewiesen werden.

Der Projektablauf wurde wie folgt definiert:

**1. Anforderungen / Requirements**

1.1. Dokumentation der Anforderungen

1.2. Pflichten und Lastenheft

**2. Spezifikation der Anforderungen**

2.1 Darstellung in Diagrammen

2.1.1 **Behaviour** Diagram

2.1.1.1 - **Use Case** Diagramm  
 2.1.1.2 - **Aktivitäts** Diagramm

2.1.2 **Structure** Diagram

2.1.2.1 - **Klassen** Diagramm  
 2.1.2.2 - **Deployment** Diagramm

2.1.3 **Strategy pattern** / design pattern (nice to have)

**3. Design / Entwickeln**

**4. Testen / Verbessern**

**5. Implementieren**

# *Spezifikation der Anforderungen*

Das Requirements Engineerig wird in zwei Teile unterteilt, die betriebswirtschaftlich zu unterscheiden sind. Es handelt sich einerseits um Stakeholders und andererseits um Shareholders. Es ist zu beachten, dass beide Gruppen eine Schnittmenge haben und daher eine Überlappung von Interessen und Bedürfnissen entsteht. Die Unterteilung erfolgt anhand des Exponierens am Markt und der Teilhabe am Risiko des investierten Kapitals.

Mit Stakeholder sind vor allem Kunden, Anleger und Abonnenten gemeint. Die Anforderungen der Kunden bestehen in erster Linie aus Informationsbedürfnissen und Renditeerwartungen.

Durch eine API-Schnittstelle wird sichergestellt, dass immer aktuelle Informationen im Tool verfügbar sind. Diese Informationen werden verarbeitet und formalisiert und anschliessen wieder an die Kunden des RTD-Tools versendet. In einer weiteren Schnittstelle werden die Transaktionsgeschäfte «Trades» in eine Plattform gestossen «gepusht» und gehandelt. Dies erfolgt in erster Linie automatisch und wird durch ein Tool namens «Tracker» analysiert. Der Tracker befindet sich ausserhalb des Tools und wird, bei Unterschreitung von den gegebenen Anforderungen, den Handel stoppen.

Shareholder sind vor allem Investoren und Anleger, die einerseits das Risiko der Gesellschaft tragen sowie eine Rendite ihres investierten Geldes erwarten. Die Unterteilung erfolgt in erster Linie an den monetären Bedürfnissen.

Alle Daten werden in einer lokalen Datenbank gespeichert und sind so beliebig für alle Klassen erreichbar und somit jederzeit für Kunden aufrufbar. Durch die stetige Verfügbarkeit der Daten für die Applikation, sind die Kunden jederzeit in der Lage Real-Time Daten über ihr Konto und Transaktionen zu erhalten. Dies widerspiegelt die Interessen der Stakeholder.

## Verhaltens-Diagramme / behaviour diagrams

### **Use Case Diagramm**

**Ein Bild, das Text, Karte enthält.

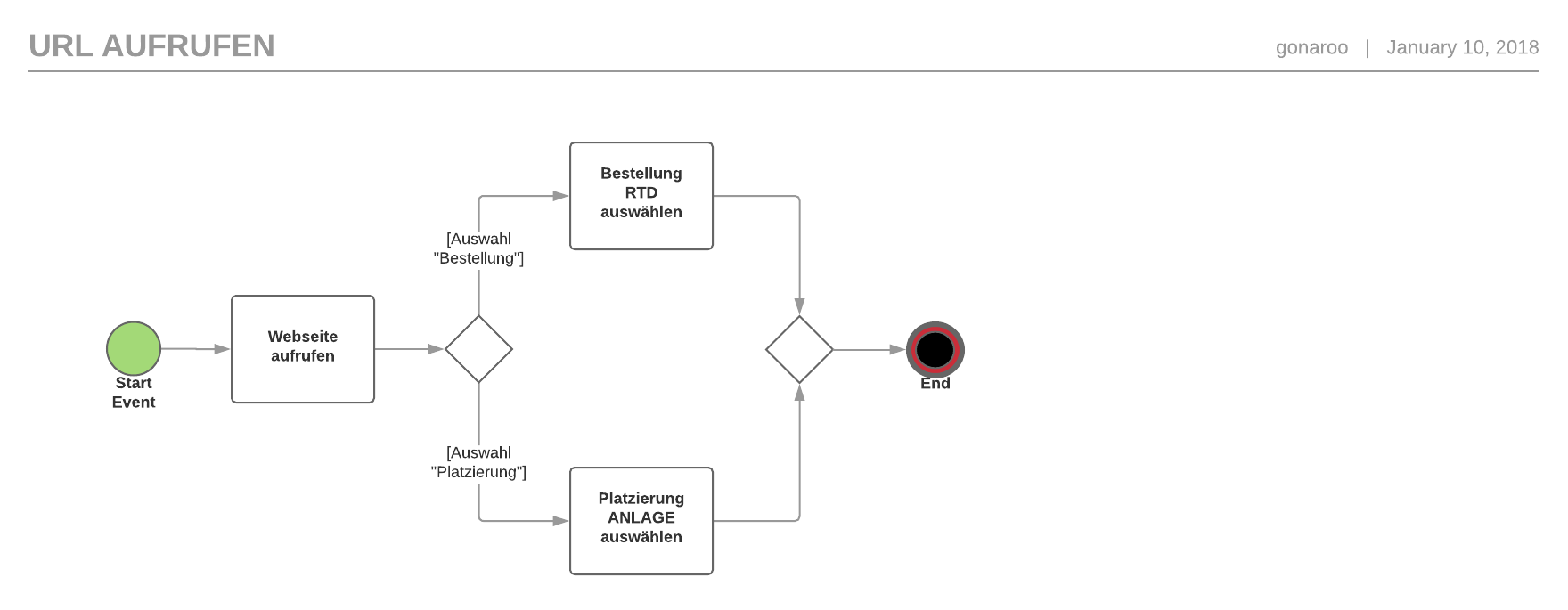
Mit sehr hoher Zuverlässigkeit generierte Beschreibung**Der Use Case, als Teil des Lastenheftes, wurde Lösungsfrei definiert und gestaltet. Das «Wie» wurde zu diesem Zeitpunkt nicht thematisiert. Diesbezüglich diente Herr Gantenbein als unser Kunde welcher später auf die Seite des Auftragnehmers wechselte.

Use Case Diagramm

### **Beschreibung Use Cases und Aktivitätsdiagramme**

Um die Use Cases verständlicher zu beschreiben, haben wir unter der Beschreibung der USE Cases das entsprechende Aktivitätsdiagramm angefügt.

#### **URL aufrufen**

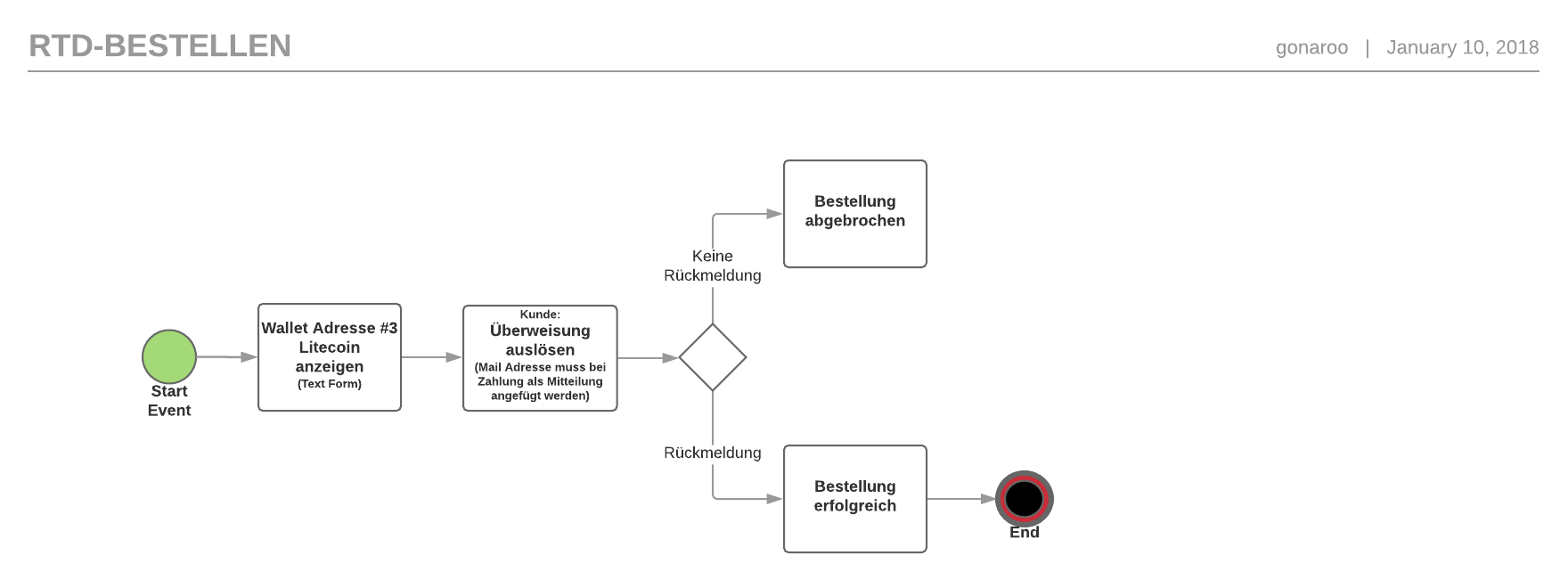
In diesem Diagramm wird der Prozess des Entscheids sowie der Platzierung der Anlage sichtbar. Hierbei haben die Anleger jeweils zwei Möglichkeiten. Es besteht einerseits die Möglichkeit den Real Time Data Report zu abonnieren sowie andererseits einen Betrag in der Anlage zu platzieren.

Aktivitätsdiagramm «URL aufrufen»

**4.1.2.2 «RTD» Real Time Data bestellen**

In diesem Szenario bestellt der Kunde eine E-Mail mit Informationen. Die Informationen beinhalten eine Anleitung, die durch verschiedene Wechselkurse Kryptowährungen profitabel gewechselt werden können. Hierbei werden die günstigsten Wechselkurse analysiert und anschliessend der «Weg» als Diagramm im E-Mail dargestellt. Hierbei ist der Zeitpunkt sowie die Berechnung als die Unique Selling Proposition zu qualifizieren.

Der Kunde bestellt via «Blockchain Transaktion» den «Email RTD»-Report. Der Preis pro «RTDR» beträgt 1 Litecoin.



Aktivitätsdiagramm «RTD bestellen»

#### **Anlage platzieren**

Der Kunde platziert Anlagen, indem er einen bestimmten Betrag in einer beliebigen Währung (Fiat oder Crypto) für das gewünschte Produkt einsetzt. Die Zuordnung geschieht automatisch, da für kurzfristige und langfristige Anlagen unterschiedliche Konten existieren.

Ein Bild, das Screenshot enthält.

Mit hoher Zuverlässigkeit generierte BeschreibungEin Bild, das Screenshot enthält.

Mit hoher Zuverlässigkeit generierte Beschreibung

Aktivitätsdiagramm «Anlage platzieren»

#### **GS Tracker / Kill-Switch**

Der GS Tracker könnte man als «Überwacher» aller Transaktionen bezeichnen.

Mit einem Kill-Switch werden bei Unterschreitung der vordefinierten Angaben alle Transaktionen unterbrochen und anschliessend wieder neu platziert.

#### **Engine Arbitrage: Algorithmus 1 – Fill The Gap**

Das ist eine langfristige Anlagestrategie, die sich durch kleines Risiko auszeichnet. Gleichzeitig muss sie langfristig gehandelt werden, da der Gewinn, durch das Exponieren der Investition am Markt, wächst.

Der Einsatz des Kunden wird für exakt 3 Monate verwendet. Dies ist der, von uns prognostizierte, Zeitraum, in welchem Risiko und Gewinn im Gleichgewicht sind. Dabei werden die Lücken zwischen Angebot und Nachfrage mit dem Algorithmus ausgenutzt, um sich einen Vorteil zu schaffen und den Erlös abzuschöpfen. Es handelt sich hier um «Kleinst-Margen», die gehandelt werden.

Funktionsweise: Algorithmus

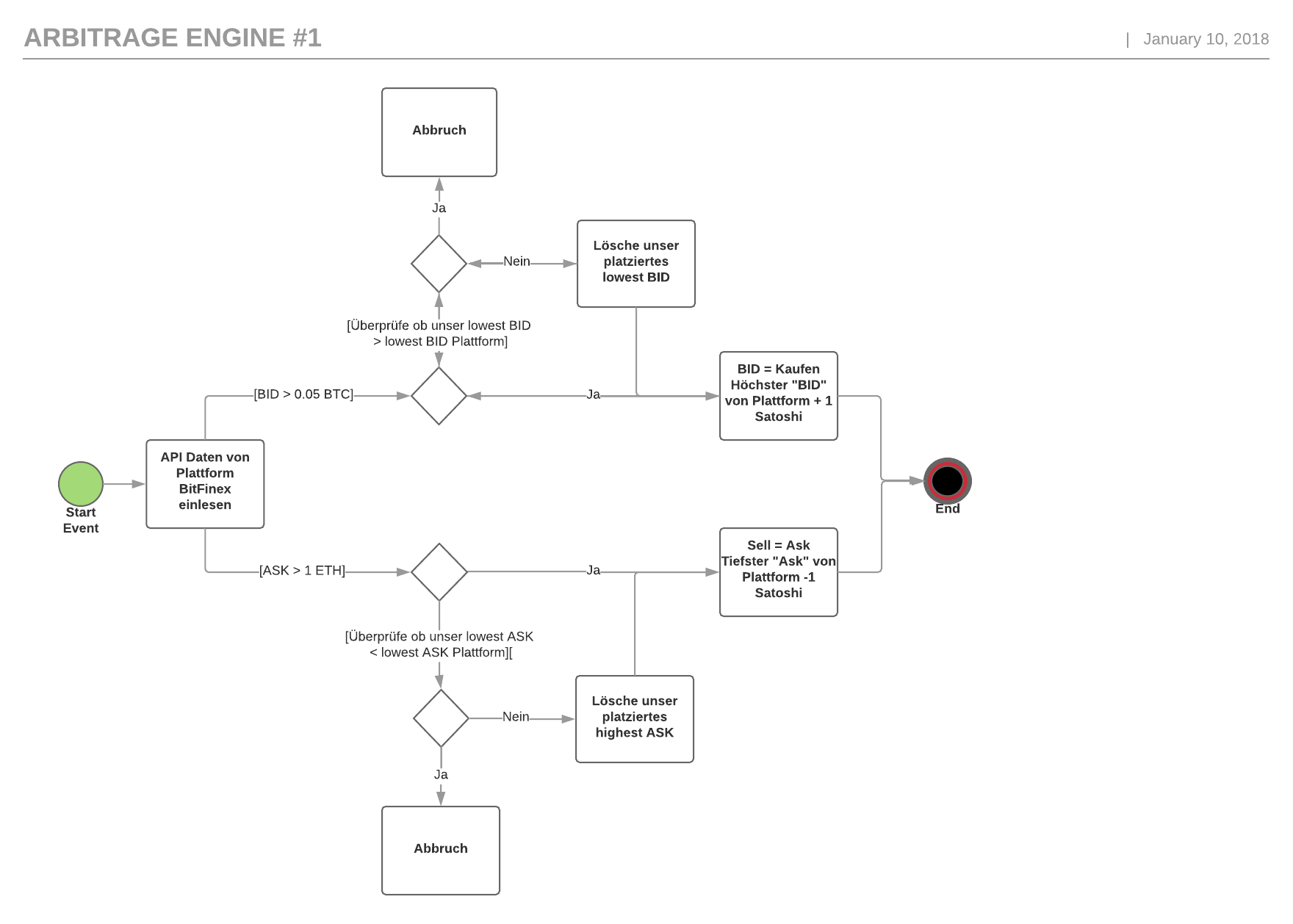
Read API – Set ORDERS  
Buy Order Input: API HTTPS Request Highest Bid plus 1 Rappen = Output x  
Sell Order Input: API HTTPS Request Lowest Sell minus 1 Rappen = Output y  
X = Buy Price Y = Sell Price | Amount = 1 ETH

Filter Highest Bid / Lowest Sell above 1 LTC volume  
Watchdog is active when SELL/BUY balance < 1 ETH  
X = Highest Bid => Still good X < Highest Bid => Cancel oldest order  
Y = Lowest Sell => Still good Y > Lowest Sell => Cancel oldest order

Die Erklärung des «Bid-Ask-Spread» Prinzips hat Investopedia.com wie folgt publiziert:

«The bid-ask spread reflects the supply and demand for an asset. The bids represent the demand, and the asks represents the supply for the asset. The depth of the bids and the asks can have a significant impact on the bid-ask spread, making it widen significantly if one outweighs the other or if both are not robust. Market makers and traders make money by exploiting the Bid-Ask Spread and the depth of bids and asks to net the spread difference.

The Bid-Ask-Spread's Relation to Liquidity: The size of the bid-ask-spread from one asset to another differs mainly because of the difference in liquidity of each asset. Certain markets are more liquid than others. For example, currency is considered the most liquid asset in the world, and the Bid-Ask Spread in the currency market is one of the smallest (one-hundredth of a percent); in other words, the spread can be measured in fractions of pennies. On the other hand, less liquid assets, such as small-cap stocks, may have spreads that are equivalent to 1 to 2% of the asset's lowest ask price. » *https://www.investopedia.com/terms/b/bid-askspread.asp*



Aktivitätsdiagramm «Engine Arbitrage»

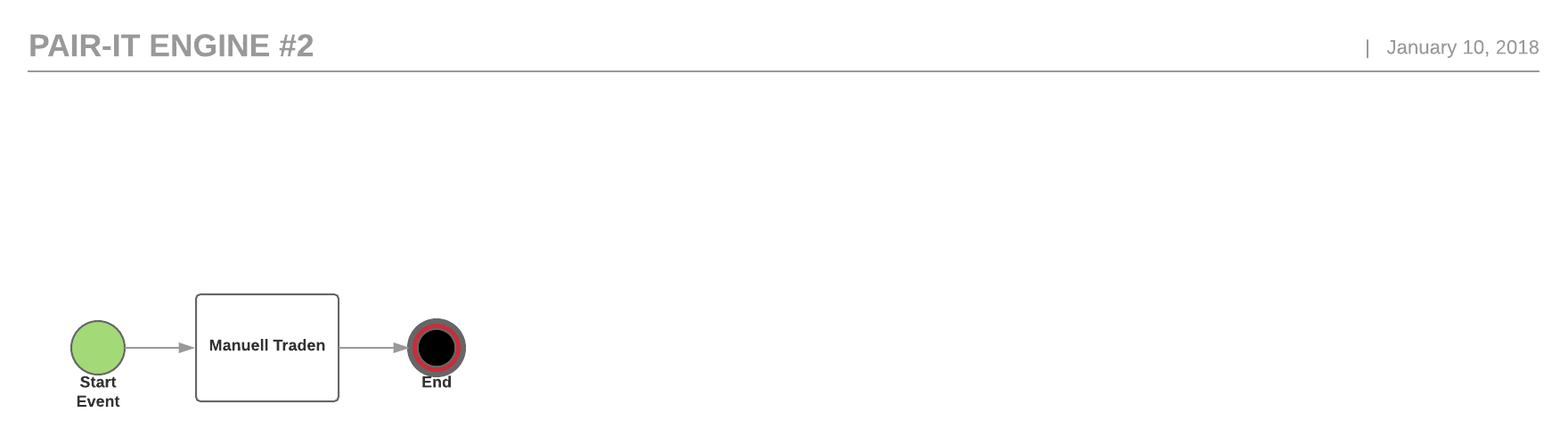
#### **Engine PaiRiT: manuell traden – Pair the Pair**

Read XML - Provide Array for RTDF

\*Still in Development\* Algorithmus 2 - Geplanter Launch: Go Live 2

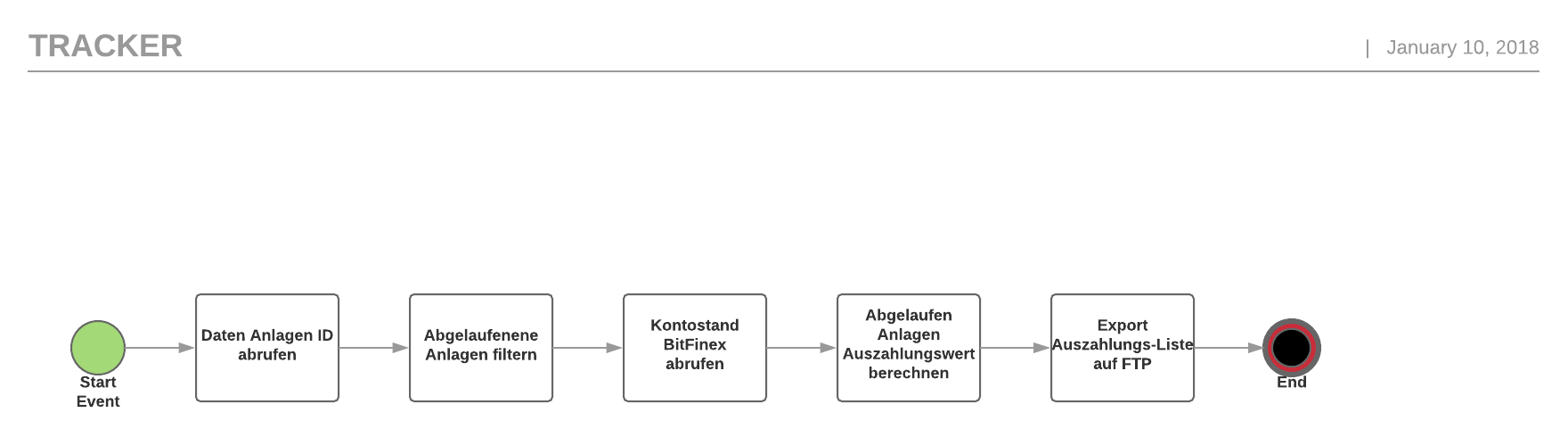
Dies ist die kurzfristige Anlagestrategie für gierige und risikobewusste Anleger. Der Einsatz wird genau 2 Wochen lang verwendet. Durch den Einsatz eines zweiten Algorithmus in Kombination mit unserem «RTD»-Tool werden die Anlagen von Hand verwaltet. Dadurch sind höhere Gewinne erzielbar. Es birgt jedoch auch höhere Risiken. Für die tägliche Betreuung der Engine werden hohe Gebühren auf den Gewinn erhoben.

Wikepadia hat die «Triangular arbitrage» wie folgt beschrieben:

«Triangular arbitrage (also referred to as cross currency arbitrage or three-point arbitrage) is the act of exploiting an arbitrage opportunity resulting from a pricing discrepancy among three different currencies in the foreign exchange market. A triangular arbitrage strategy involves three trades, exchanging the initial currency for a second, the second currency for a third, and the third currency for the initial. During the second trade, the arbitrageur locks in a zero-risk profit from the discrepancy that exists when the market cross exchange rate is not aligned with the implicit cross exchange rate. A profitable trade is only possible if there exist market imperfections. Profitable triangular arbitrage is very rarely possible because when such opportunities arise, traders execute trades that take advantage of the imperfections and prices adjust up or down until the opportunity disappears. » //Wikipedia

Aktivitätsdiagramm «Engine Pair-It»

#### **Tracker auszahlen**

Der Auszahlungsprozess erfolgt automatisch, sobald der definierte Zeitraum überschritten ist. Hierbei werden Marge, Prozesskosten abgezogen und der Gewinn anschliessend dem Kunden ausbezahlt

Aktivitätsdiagramm «Tracker auszahlen»

#### **Engine Delta Pips: Algorithmus #3 / Bonus**

\*Still in Development\*

Wertvermehrung für kurzfristige Anlagen in der semiautomatischen «Engine». Fix eingerichtete Berechnungen zu Kurvenprognosen und ausnutzen von Standardabweichungen.

## Struktur-Diagramm / structure diagram

### **Klassen Diagramm**

Das Klassendiagramm ist das mit Abstand wichtigste Strukturdiagramm der UML, wenn nicht sogar der wichtigste Diagrammtyp überhaupt. Es stellt Zusammenhänge zwischen Klassen als auch den Aufbau von Klassen dar. Da Klassen das entscheidende Merkmal objektorientierter Programmiersprachen sind, können sich Programmierer, die bisher wenig mit Softwaremodellierung zu tun hatten, sehr schnell mit diesem Diagrammtyp anfreunden. Das, was man vom Quellcode her kennt, kommt dem, was das Klassendiagramm zeigt, am nächsten. *[Aufruf am 12.12.2017]* [*http://www.highscore.de/uml/klassendiagramm.html*](http://www.highscore.de/uml/klassendiagramm.html)

Das nachfolgend aufgezeigte Klassendiagramm bildete ursprünglich die Basis für unsere Klassen in «Java». Während der Umsetzung resp. beim Erstellen des Quellcodes in «Eclipse» wurde bemerkt, dass einige Klassen so nicht umgesetzt werden konnten. Danach wurden die notwendigen Anpassungen direkt im Quellcode vorgenommen. Auf eine nachträgliche Korrektur unseres ursprünglichen Klassendiagramms wurde bewusst verzichtet.

Ein Bild, das Screenshot enthält.

Mit hoher Zuverlässigkeit generierte Beschreibung

Klassendiagramm

### **Verteilungsdiagramm / Deployment Diagram**

Das Verteilungsdiagramm (Deployment-Diagramme) stellt dar, wie das System auf der vorhandenen Hardware und der vorhandenen Netzwerktopologie verteilt ist und wie die Kommunikationsbeziehungen aussehen.

*[Aufruf am 12.12.2017]* [*http://www.software-kompetenz.de/servlet/is/27032/?print=true*](http://www.software-kompetenz.de/servlet/is/27032/?print=true)

Um das Projekt termingerecht realisieren zu können, wurde auf das Erstellen eines Deployment Diagrams verzichtet.

# *Technische Umsetzung*

## Design / Entwicklung

Anhand des Klassendiagramms wurden die entsprechenden Klassen Schritt für Schritt in Java geschrieben und laufend überprüft. Wie bereits oben erwähnt, mussten einige Klassen anders als im ursprünglichen Klassendiagramm umgesetzt werden. Viele Klassen und Schnittstellen ergeben sich im Lauf der Programmierung. Geplante Features sind aufgrund technischer Limitationen teilweise nicht möglich in Java oder vom Blockchain-Funktionsangebot nicht verfügbar. Auf die nachträgliche Anpassung der einzelnen Klassen im Klassendiagramm wurde bewusst verzichtet, da dieses Diagramm als «Tool» verwendet wurde und nicht für einen weiteren zukünftigen Gebrauch gedacht ist.

## Datenbank / Plattform

Damit unser Programm mit aktuellen «Börsen»-Daten überhaut arbeiten kann, resp. funktioniert, müssen spezielle Verknüpfungen/Schnittstellen mittels APIs hergestellt werden.

Man entschied sich die Schnittstelle über einen separaten Server laufen zu lassen. Mit dem Node.Js fand man eine Plattform, bei der beliebig viele APIs kreiert werden können und welche zusätzlich über ein User-Management verfügt.

Mit Node, Loopback, MongoDB und Loopback-Mongo-Connector wurden 4 NPM-Installations-Pakete installiert. Danach konnten mit Hilfe von Loopback mehrere schöne Swagger-API’s gebaut werden, welche als gewünschten REST-API’s dienen. Die Node stellt die API’s parat. Sämtliche Daten werden nicht im Memory, sondern in einer lokalen Mongo-Server-Datenbank gespeichert.

Java greift auf die Daten über den Localhost zu. Da die Verbindung lokal funktioniert, ist nun die Implementation auf einem Live-Linux-Server ebenso möglich.

Neue Properties können über CLI (Command Line Interface) oder auch Bearbeiten der Datenbank.json Datei angelegt werden.

Der Prozess wird beschrieben in: DB Installation Steps.txt

## Testen / Verbessern / Implementieren

Nach dem Schreiben der Klassen in JAVA wurden diese getestet, korrigiert und verbessert.

Folgende Klassen können heute erfolgreich ausgeführt werden:

* src.Interface.java
* src.user.User.java
* src.db.hsqldb.ConnectHSQLDB.java
* lib.hsqldb.jar
* run.Server.bat (wird benötigt HSQLDB)

Bei der Implementierung hat sich gezeigt, dass die angedachte Lösung, die hsqldb.jar als «Standalone»-Datenbank mit dem Eclipse HSQL Database Manager zu integrieren, nicht umgesetzt werden konnte. Als alternative Lösung wurde ein lokaler HSQLDB-Server eingerichtet.

Für die weitere Entwicklung sind verschiedene Libraries nötig, welche erfolgreich implementiert werden konnten:

* org.springframework.com
* org.apache.log4j
* com.google.common
* com.google.thirdparty
* javax.mail.jar

Der Code ist nun bereit, den User ans Datenbankmanagement anzubinden, sowie die Anlagen Klasse mit den API’s zu verbinden und die Strategie umzusetzen.

# *Zusammenfassung / Schlusswort*

Bei der Umsetzung unseres etwas komplexen Auftrags, erkannten wir, wie wichtig eine einheitliche Sprache für das Verständnis sowie die Implementierung eines neuen Programmes sind. Die UML Methoden waren für unser Projekt äusserst hilfreich und halfen diverse Fehler und Lehrläufe zu vermeiden.

Wir haben mit der Umsetzung dieses Projektes Erfahrungen gesammelt, welche uns für künftige Projekte sehr dienlich sein werden. Natürlich sind wir auch stolz darauf ein Programm entwickelt zu haben, welches bald erfolgreich eingesetzt werden kann.

# *Glossar*

Um Fehler bei Übersetzungen zu vermeiden sowie im Sinne der Urheber zu handeln, wurden die Definitionen im Glossar in der ursprünglichen Sprache der Quelle belassen. Wir haben ausschliesslich Quellen in den Sprachen Englisch und Deutsch verwendet.

Address

A Bitcoin address looks like a long string of random letters and numbers. Addresses are used when you want to send or receive Bitcoins. Each wallet has at least one receiving and one sending address. When you want to receive Bitcoins to your wallet you need to give the sender your receiving wallet address.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

AML and KYC

Anti-Money Laundering and Know Your Customer is a group of laws in the United States that require Bitcoin sellers to know who their customers are. Certain sellers will ask for your identification to comply with these laws. Other countries around the world may have similar laws and requirements.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

API

API ist die Abkürzung für «Application Programming Interface». Ein API ist also eine Schnittstelle: Es verbindet Soft- und Hardwarekomponenten, beispielsweise Anwendungen, Festplatten oder Benutzeroberflächen

*[Aufruf am 12.12.2017]* [*https://praxistipps.chip.de/was-ist-api-einfach-erklaert\_41370*](https://praxistipps.chip.de/was-ist-api-einfach-erklaert_41370)

Ashdrake’d

There once was a trader who went by the handle Lord Ashdrake. He was a Romanian programmer, and was a prolific force during the nuclear Bitcoin winter in 2014 and 2015. His skill was shorting Bitcoin, and that strategy worked like a charm until it didn’t.

When Bitcoin finally broke and held \$300, Ashdrake performed his usual action of shorting Bitcoin. Unfortunately this time, the price continued through \$300 to \$500, and almost touched \$600 in under 2 months.

He completely blew up his account to the point where he could no longer trade Bitcoin. His folly was being unable to shift into a bull market mindset. The trader community coined the term “to be Ashdrake’d.” It meant to completely blow up your trading account by shorting Bitcoin.

Leading up to Monday’s CBOE Bitcoin futures launch, the financial media constantly droned on that institutional investors would line up to short Bitcoin into the ground. However within twelve hours after the launch, the CBOE Jan Bitcoin future hit the circuit breaker three times, and was up over 20\%.

Interactive Brokers was so afraid of being Ashdrake’d they did not allow clients to go net short the futures contract. They have since reversed that stance, but shorts must post a whopping 400\% margin.

*[Aufruf am 12.12.2017]* [*https://hash.bg/forum/topic/1253-bitmex-crypto-trader-digest/*](https://hash.bg/forum/topic/1253-bitmex-crypto-trader-digest/)

Blockchain

The Blockchain is the technology behind Bitcoins. It’s what makes it work. Bitcoin transactions are sent to the Blockchain so that miners can put them into the blocks that they mine. Once a block containing your transaction is mined it has been added to the Blockchain and your transaction receives one confirmation.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Blockchain.info

Not to be confused with the Blockchain, blockchain.info is one of many websites that work as Blockchain explorers. These websites allow you to ‘see’ the Blockchain and what’s going on in it. In practical terms this lets you view your transactions and see if they are confirmed or not. You can also check to see how many unconfirmed transactions there are, the more the longer it will take for your transaction to become confirmed.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Blockchain Transaktion

Jede Transaktion wird in einem Block auf der Blockchain gespeichert und kann optional eine Nachricht enthalten. Bei einer Transaktion wird eine Nachricht hinzugefügt, damit der Admin weiss, wohin er die E-Mail zustellen soll.

*Quelle: Michael Gantenbein*

Bitcoin Block

A Bitcoin block is one “link” in the Blockchain. One block contains a group of Bitcoin transactions that have been confirmed. When Bitcoin miners ‘mine’ these blocks they calculate through computer algorithms which when put in the Blockchain, confirms transactions. In turn miners receive transaction fees and newly created Bitcoins for confirming transactions.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

BTC/Coin

BTC is the acronym for Bitcoin. Coin, coins et. al. are also slang terms for Bitcoin.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Circle/Coinbase

Two Bitcoin exchanges based in the U.S. that require you to verify your identity before being allowed to trade, along with other limitations.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

CLI

Als Command Line Interface ([CLI](https://www.it-visions.de/glossar/alle/430/Common_Language_Infrastructure.aspx)) wird eine Mensch-Maschine-Schnittstelle bezeichnet, bei der die Anweisungen an die Maschine durch Eingabe von Befehlswörtern erfolgt.

*[Aufruf am 12.12.2017]* [*https://www.it-visions.de/glossar/alle/3268/Command\_Line\_Interface.aspx*](https://www.it-visions.de/glossar/alle/3268/Command_Line_Interface.aspx)

Cold storage/Cold wallet

This refers to a way of storing Bitcoins in a safe and secure way, offline. This is also how the clear majority of Bitcoins LocalBitcoins hold on to are stored.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Confirmation

For a Bitcoin transaction to be completed it needs to be confirmed. Good habits are to wait for at least 3-6 confirmations before you can consider a transaction good. A confirmation happens when a transaction has been added to a block that Bitcoin miners successfully mine.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Exchange

An exchange is a platform/service where users can change one type of currency for another.

*[Aufruf am 12.12.2017]* [*https://bitseller.biz/en/Faq*](https://bitseller.biz/en/Faq)

Fiat

Fiat comes from Latin and means “let it be done” or “it shall be”. It’s used as a term to mean all currencies that derive their value from governmental regulation or other central authorities.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Hot wallet

The opposite of a cold wallet / cold storage. A hot wallet is located on a computer connected to the internet. At LocalBitcoins the hot wallet is where we store a small number of Bitcoins at a time. It’s from here where your transactions are sent. We only keep a very small amount of Bitcoin here to protect ourselves from hackers.

*[Aufruf am 12.12.2017]* [*https://localbitcoins.com/faq*](https://localbitcoins.com/faq)

Mining

This is what Bitcoin Miners do to confirm transactions and add them to the Blockchain. When a Bitcoin miner mines, he uses computers to do difficult calculations which answers are easily proven to be correct. This way, when a miner completes a calculation anyone can easily see that the miner has done work. This work can then be used to create a Bitcoin block. What miners get in return from mining is new Bitcoins and transaction fees.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Multisig

Multisignature, often heard in combination with wallet. A multisignature wallet is one that has several cryptographic keys concerned with it. This way if you have two of three keys, say, you can move money out of it. But with only one you cannot do anything.

*[Aufruf am 12.12.2017]* [*https://localbitcoins.com/faq*](https://localbitcoins.com/faq)

P2P

Peer-to-Peer. It’s a form of network structure. Rather than having everything go via a centralized node, in a peer-to-peer structure everything goes from user to user. There’s no centralized hub to rely on.

*[Aufruf am 12.12.2017]*

*<https://localbitcoins.com/faq>*

Satoshi

A Satoshi is the smallest amount of Bitcoin that can be sent, or 0.00000001 BTC.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Satoshi Nakamoto

The pseudonym of the person who invented Bitcoin. No one knows who he is, but he does own quite a nice amount of BTC.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

SEPA

Single Euro Payment Area, SEPA transfers are a special kind of bank transfers within the EU.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Transaction

A Bitcoin transaction, the actual act of moving Bitcoin from one wallet to another.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Transaction fee

Since the Bitcoin miners work to confirm your transactions, they won’t do this for free. Rather you pay a tiny amount of Bitcoin to them to help them continue to confirm. Therefore each time you move Bitcoin, it costs a tiny bit.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

txid

Transaction Id. This is a long string of numbers and letters that is the ID of a single transaction. It’s handy to know as you can paste this into a Blockchain explorer and see what’s up with your transaction. It’s mainly used to see how many confirmations a transaction has.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

Wallet

The place where you store your Bitcoins. This can be quite confusing to new users, but a wallet is essentially just a Bitcoin address. A Bitcoin wallet can exist on a service such as LocalBitcoins, in an app on your phone or computer, or even on a piece of paper! As it’s just a series of numbers and letters it does not matter where it’s stored. It’s completely OK to use your LocalBitcoins account as your wallet when you’re still new. But when you start to accumulate more and more Bitcoin you may want to think of different ways to store them.

*[Aufruf am 12.12.2017]* [*https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/*](https://instalexeng.userecho.com/forums/2-knowledge-base/topics/213-cryptocurrency-glossary/)

# *Quellenverzeichnis*

Sämtliche Angaben der Quellen wurden im Text integriert.