Grundlagen und Anwendungsmöglichkeiten der Blockchain-Technologien

FOM Hochschule für Oekonomie & Management, Nürnberg

Dr. Peter Vatter

16.05.2018

Agenda

1. Grundlagen

1.1 Begriffe: Bitcoin, Blockchain, Distributed Consensus

1.2 Konsens in verteilten Systemen: die Byzantinischen Generäle

1.3 Hash-Funktionen und Hash-Pointer

1.4 Asymmetrische Kryptographie

1.5 Spieltheoretische Aspekte

2. Aspekte in der Anwendung

2.1 Skalierbarkeit

2.2 Verteilte Datenhaltung

2.3 Smart Contracts

3. Use Cases

3.1 Supply Chain & Logistics

3.2 Governance

3.3 Automated Services

3.4 Finance

4. Ausblick

5. Q & A

1. Grundlagen

1.1 Begriffe: Bitcoin, Blockchain, Distributed Consensus

...

1.1 Begriffe: Bitcoin, Blockchain, Distributed Consensus Technologies

Bitcoin

- Ist eine Kryptowährung und die bekannteste Anwendung der Blockchain-Technologie
- Beruht auf dem Teilgebiet der Informatik der „verteilten Systeme“

Verteilte Systeme

- sind Systeme, bei denen es keine zentrale Instanz gibt, der alle Teilnehmern vertrauen müssen

Blockchain

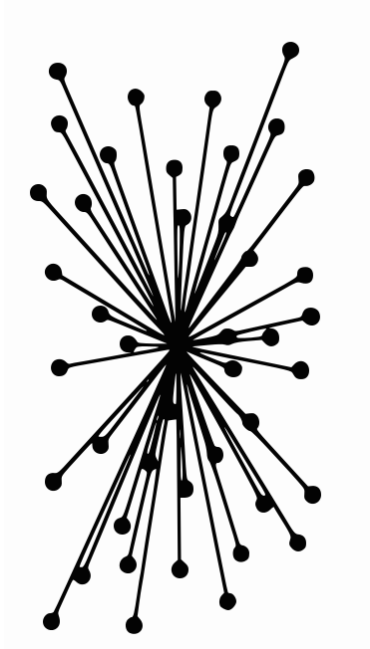
- Ist eine Form, in einem verteilten System Informationen für alle transparent und nachvollziehbar zu speichern.
- Speichert neue Informationen in einer fortlaufenden Kette an „Blocks“

Distributed Consensus Technologies

- Wäre der bessere Fachbegriff, da es auch „Blockchains“ ohne „Blocks“ und ohne „Chain“ gibt

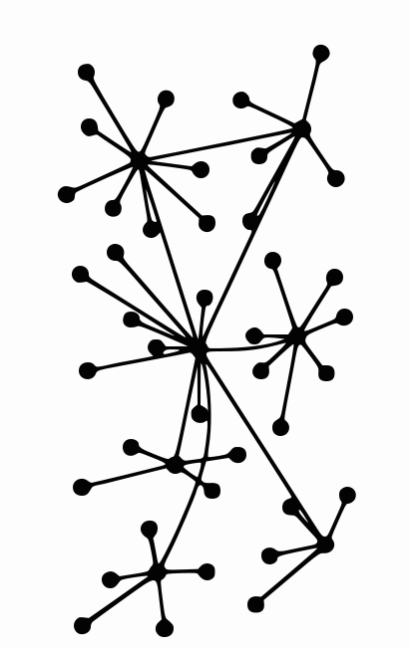
1.1 Begriffe: Bitcoin, Blockchain, Distributed Consensus Technologies

Zentrale Systeme



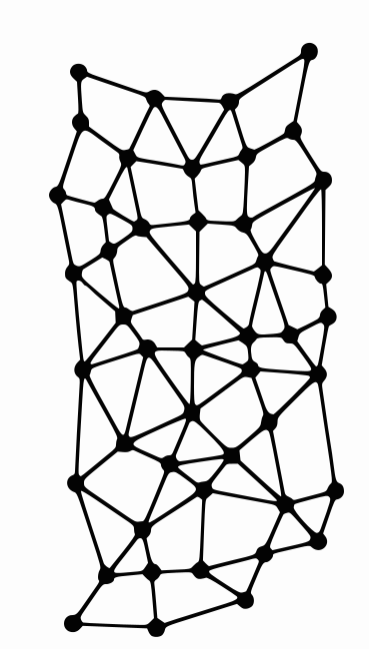
- Es gibt eine zentrale Instanz
- Vertrauen?
- Engpass?

Dezentrale Systeme



- Es gibt mehrere Instanzen
- Vertrauen?
- Engpässe besser beherrschbar

Verteilte Systeme



- Alle Teilnehmer sind gleich
- ausfallredundant
- selbstregulierend

Quelle: commons.wikimedia.org

Anforderungen an verteilte Systeme

- ausfallsicher sein — Teilnehmer können ausfallen / verschwinden
- vertrauenswürdig sein — Teilnehmer können böser Absicht sein
- skalierbar sein — Teilnehmer können in unbekannter Zahl dazu kommen

- **müssen handlungsfähig, d.h. in der Lage sein, einen Konsens zu finden!**

1. Grundlagen

...

1.2 Konsens in verteilten Systemen: die Byzantinischen Generäle

...

1.2 Konsens in verteilten Systemen: die Byzantinischen Generäle

Wie kommen alle Teilnehmer eines verteilten Systems zu einem Konsens?

- Abstimmung nicht möglich
  - Kein Teilnehmer kann sicher sein, dass er alle anderen Teilnehmer kennt (tut er meist nicht)
  - Ein Teilnehmer kann sich als mehrere ausgeben („Ich bin Brian; und meine Frau ist auch Brian“)
  - Keine zentrale Instanz, welche die „Stimmen auszählt“

Konsensfindung ist im Verteilten System zunächst nicht möglich.

Es funktioniert in der Praxis aber besser als in der Theorie.

1.2 Konsens in verteilten Systemen: die Byzantinischen Generäle

Das Modell

- Der Sultan hat seinen Generälen befohlen, die Stadt Byzanz einzunehmen
- Die Divisionen sind weit um die Stadt verteilt
- Ziel ist es, sich auf einen Zeitpunkt für den gleichzeitigen Angriff zu verständigen
- Die Generäle kommunizieren über Boten mit anderen Divisionen

Jeder einzelne General weiß nicht, ob ...

- andere Divisionen noch existieren oder schon geschlagen wurden
- die Botschaften anderer Generäle valide sind
- die anderen Generäle kooperieren

1. Grundlagen

...

1.3 Hash-Funktionen und Hash-Pointer

...

Hash Funktion H(x)

- Ein Datum x wird durch eine mathematische Funktion unkenntlich, aber wiedererkennbar gemacht („Einwegfunktion“)

  H: x → H(x) x kann beliebig lang sein, H(x) hat feste Länge (z.B. 256 Bit)

- Schlechtes Beispiel: Quersumme

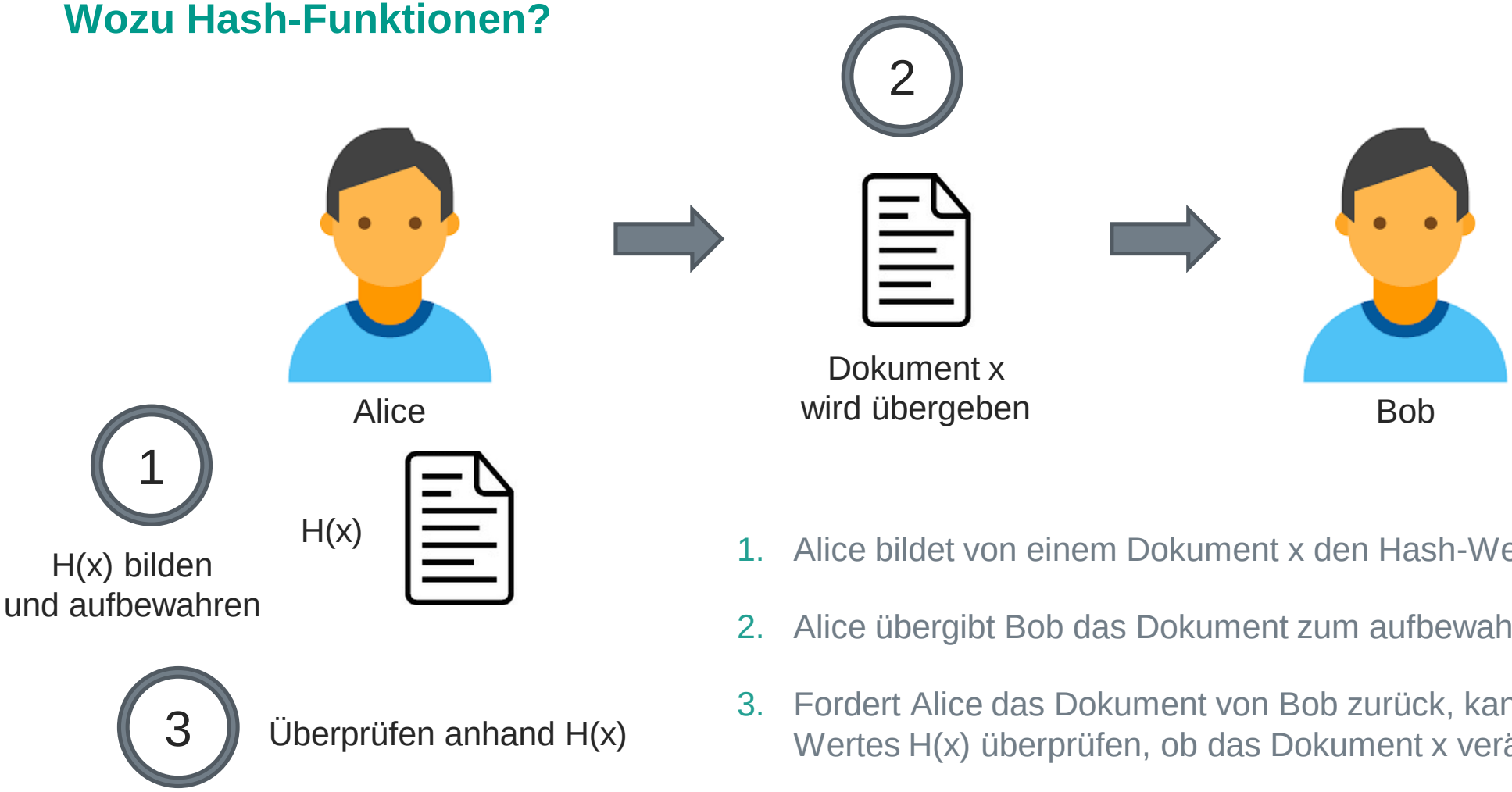
  1337 → 14

Anforderungen

Eindeutigkeit:	Aus identischem x muss immer identisches H(x) werden.
Irreversibilität:	Aus H(x) darf x nicht berechnet werden können.
Kollisionsresistenz:	Es dürfen keine x und x‘ findbar sein, welche H(x) = H(x‘) haben.

1.3 Hash-Funktionen und Hash-Pointer

Wozu Hash-Funktionen?



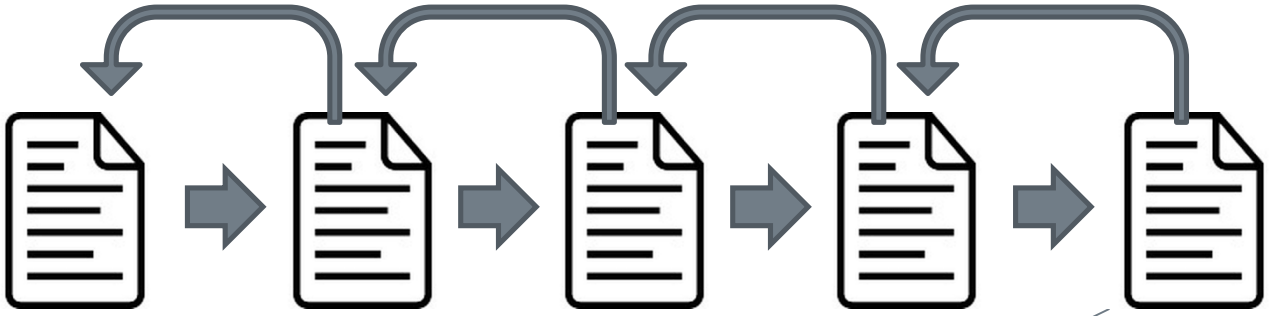
1. Alice bildet von einem Dokument x den Hash-Wert H(x)
2. Alice übergibt Bob das Dokument zum aufbewahren / speichern
3. Fordert Alice das Dokument von Bob zurück, kann sie anhand des Hash-Wertes H(x) überprüfen, ob das Dokument x verändert wurde.

1.3 Hash-Funktionen und Hash-Pointer

Hash-Pointer

- Hash-Wert + Speicheradresse des Dokuments → Hash-Pointer

Blockchain



- „Alte“ Blocke können nicht mehr unerkannt verändert werden
- Zum Verändern eines einzigen Datums muss die gesamte Blockchain neu erstellt werden
- Regel: Alle Clients richten sich nach der längsten Kette
- Wenn das Berechnen von Blöcken „teuer“ ist, sind alte Blöcke gut gegen Manipulation geschützt

Blockheader

Block-ID (Adresse)
Hash-Pointer des vorhergehende Blocks
Block-ID des vorhergehenden Blocks
Datum

Zwischenfazit

- Somit ist das Problem gelöst, wie ein Verändern bestehender Datenbestände unterbunden werden kann

Aber, es ist immer noch unklar

- ... wie neue Blöcke angehängt werden?
- ... wer einen neuen Block generieren darf?
- ... wie sichergestellt wird, dass die Informationen im neuen Block valide sind?

1. Grundlagen

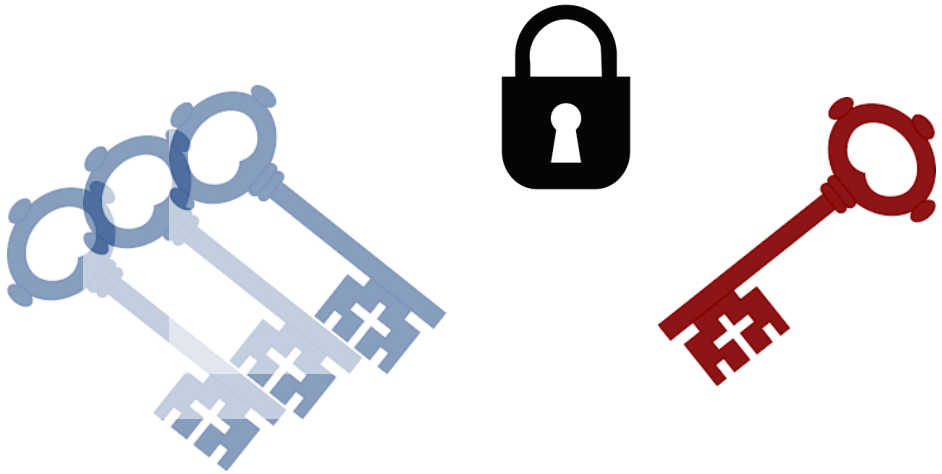
...

1.4 Asymmetrische Kryptographie

...

1.4 Asymmetrische Kryptographie

- Bei der asymmetrischen Verschlüsselung gibt es zwei Schlüssel



Schlüssel P
(public key)

Schlüssel S
(secret key)

1. Wird das Schloss mit einem Schlüssel (1) verschlossen, kann es NUR mit Schlüssel (2) geöffnet werden
2. Schlüssel P wird vervielfältigt und veröffentlicht (Public key)
3. Schlüssel S wird geheim gehalten (Secret key)

1.4 Asymmetrische Kryptographie

- **Verschlüsseln**

Schlüssel P
(public key)

1. Dokument wird mit dem Public Key *P* verschlüsselt
2. Verschlüsseltes Dokument kann nur mit dem Secret Key *S* entschlüsselt und gelesen werden, also nur vom Eigentümer des Schlüssels

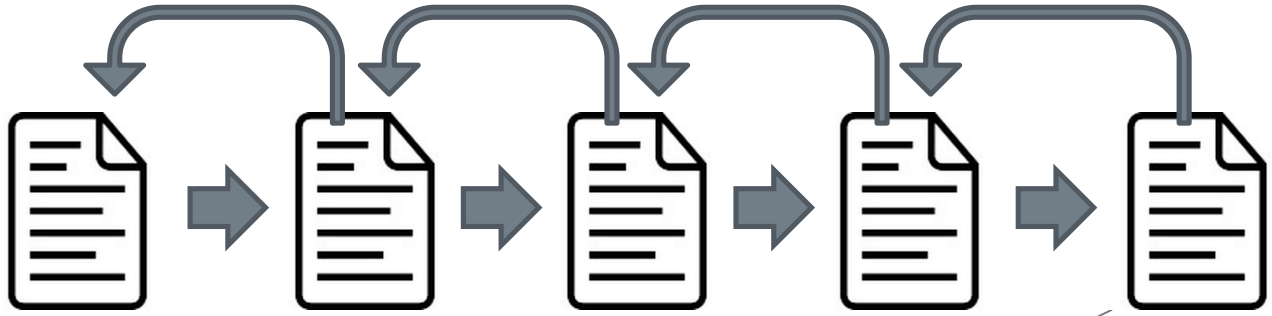
- **Signieren**

Schlüssel S
(secret key)

1. Dokument wird vom Eigentümer des Schlüsselpaars mit dem Secret Key *S* verschlüsselt
2. Verschlüsseltes Dokument kann von allen mit dem Public Key *P* gelesen werden
3. Es ist außerdem sichergestellt, dass das Dokument vom Eigentümer des Schlüsselpaares stammt

1.4 Asymmetrische Kryptographie

Aufbau einer Transaktion



1. Blocks enthalten Transaktionen
2. Transaktionen müssen vom Sender signiert sein
3. Sobald ein Block erzeugt ist, wird er im Netzwerk verteilt
4. Jeder Client prüft den Block formal
5. Entspricht der Block den Kriterien gilt er als akzeptiert

Transaktion

mehrfach bis Block-größe erreicht

Von Konto-ID1 (Sender)
Nach Konto-ID2 (Empfänger)
Betrag
Neuer Kontostand ID1
Neuer Kontostand ID2
Signatur ID1 (Sender)

1. Grundlagen

...

1.5 Spieltheoretische Aspekte

...

Wie bringt man die Teilnehmer dazu, sich fair zu verhalten?

- Entscheidend ist u.a. das Anhängen neuer Blocks an die existierende Kette (Mining)
- Leider keine Möglichkeit: Zufälliges Auswählen (nicht alle Teilnehmer bekannt)

Mining und Proof-of-work

- Das Anhängen eines neuen Blocks darf nicht zu einfach sein (sonst würden alle ständig neue Blockchain-Fragemente bilden)
- Das Recht, einen Block zu generieren ist an eine nicht-monopolisierbare Ressource geknüpft (oft Rechenleistung)
- Es muss ein aufwändiges Hash-Puzzle gelöst werden (Proof-of-work)
- Als Belohnung darf der Miner eine Überweisung „aus dem Nichts“ an sich selbst einfügen („Block Reward“)

- Die Clients akzeptieren den Block, wenn er
  - formal korrekt ist
  - Teil der längsten Kette ist

* Aktuell wird angestrebt, den Proof-of-work zu ersetzen (Stromverbrauch)

2. Aspekte in der Anwendung

2.1 Smart Contracts

2.2 Skalierbarkeit

2.3 Verteilte Datenhaltung

Was kann alles „Konsens“ sein?

- Jede Art von Daten / Informationen
- Im Fall von Kryptowährungen: Transaktionen und Kontostände
- Bei Informationen (Problem der Incentivierung, aber lösbar)
- Bedingte Transaktionen / Informationen

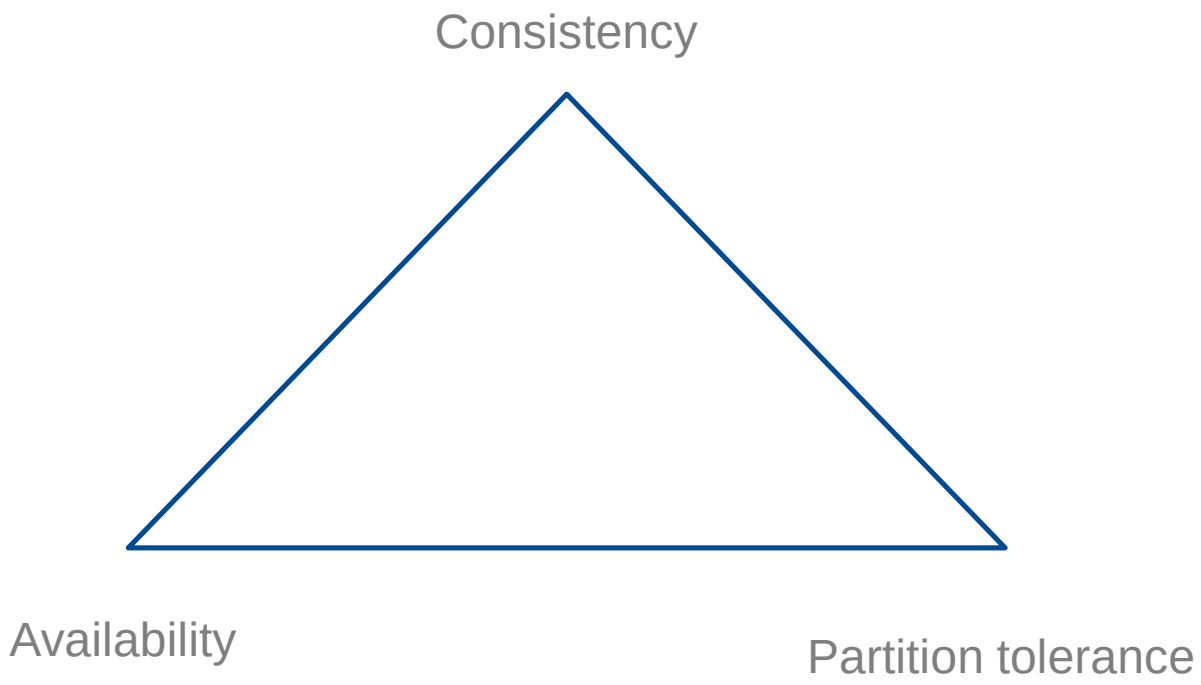
a) Überweise von *Alice* an *Bob* Betrag *x*, WENN *Bedingung* erfüllt ist!

- Durchführbar mit allen Bedingungen, die auf verfügbaren und verlässlichen Informationen beruhen
- Gut, wenn diese Informationen auch in der Blockchain gespeichert sind -> verfügbar, verlässlich
- Blockchain wird so zur „State Machine“

b) Ändere *Datum x* zu *Wert y*, WENN *Bedingung* erfüllt ist!

- „Next big thing“, umgesetzt z.B. in Ethereum

Das CAP-Theorem



- **Consistency:** Alle Replikate eines Datensatzes müssen konsistent sein
- **Availability:** Das System soll im Sinne akzeptabler Antwortzeiten verfügbar sein
- **Partition tolerance:** Das System soll auch bei Ausfall von Knoten oder Verlust von Nachrichten sinnvoll arbeiten

- Es können immer nur zwei der drei Eigenschaften gewährleistet werden!

2.3 Verteilte Datenhaltung

Wem obliegt die Data Ownership?

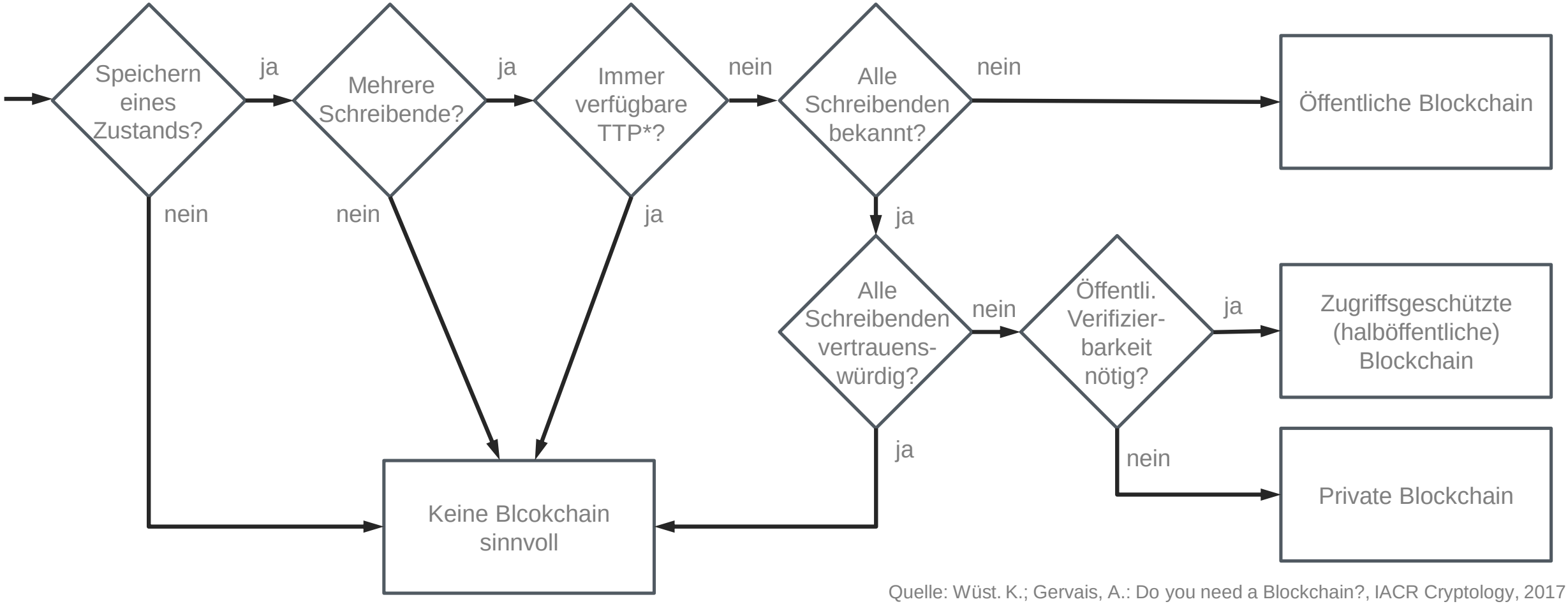
- Bei Beteiligung mehrerer Stakeholder stellt die Frage nach der Data Ownership oft ein entscheidendes Hindernis dar.
- Data Owner muss Vertrauen aller Teilnehmer genießen
- Ist dies nicht lösbar → Blockchain

Stufen der Zugreifbarkeit

- Öffentliche Blockchain (Kryptowährungen)
- Halböffentliche / Konsortiumsblockchain
- Private Blockchain

Quelle: www.smartdatacolective.com

Brauche ich überhaupt eine Blockchain?



Quelle: Wüst. K.; Gervais, A.: Do you need a Blockchain?, IACR Cryptology, 2017

3. Use Cases

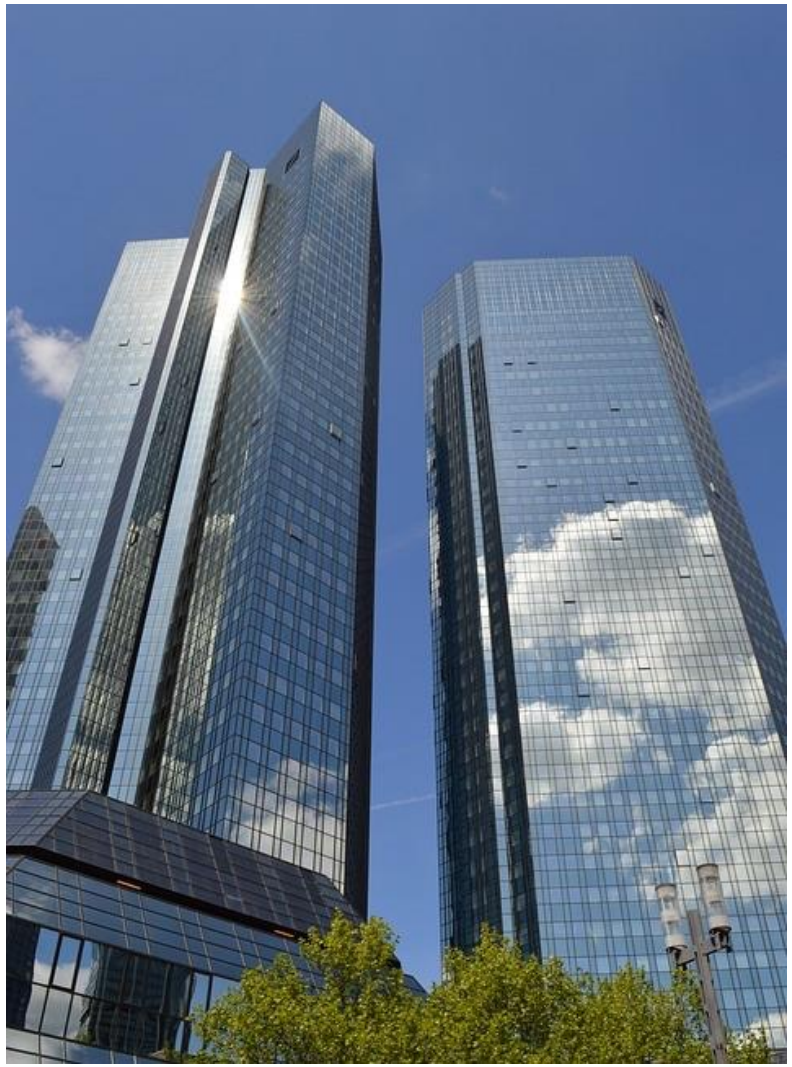
3.1 Finance

3.2 Supply Chain & Logistics

3.3 Mikroservices

3.4 Governance

3.1 Finance – Banktransaktionen



Ausgangssituation

- Zahlungsverkehr im konventionellen Bankenwesen bedeutet einen immensen Aufwand
- Auslandsüberweisungen erfordern noch händische Arbeit!!
- Dauer einer Überweisung mehrere Tage

Zielbild

- Überweisungen egal wohin, egal wie hoch der Betrag in Echtzeit (wenige Minuten) mit geringsten Kosten

Lösungsgedanke

- Durchführung der Transaktionen mittels Kryptowährungen (z.B. Ripple)

3.1 Finance – Funding

Ausgangssituation

- Handeln von Wertpapieren am Aktienmarkt erfordert sowohl von Ausgeber, als auch vom Anleger einen hohen Aufwand
- Allein die Beschaffung von Kapital ist ein Hindernis für Investitionen
- Ohne Unterstützung durch eine Bank nicht möglich

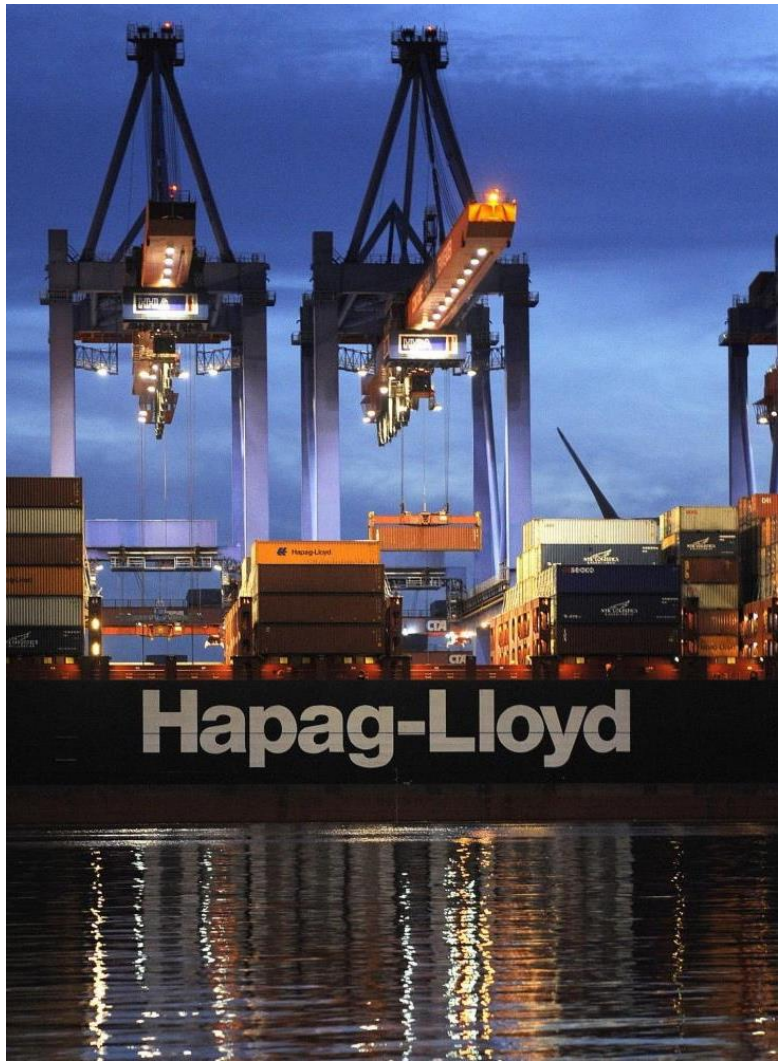
Zielbild

- Assets sollen mit minimalem Aufwand nachvollziehbar und verbindlich handelbar sein

Lösungsgedanke

- Investment in ein Unternehmen wird per Krptowährung in der Blockchain festgehalten
- Im Gegenzug wird dem Käufer ein Token (= Share) in der Blockchain übereignet
- z.B. Initial Coin Offerings (ICO)

3.2 Supply Chain & Logistics – eTraceability



Ausgangssituation

- Komplexität der Wertschöpfungsketten steigt zunehmend
- China fordert vollständige bidirektionale Rückverfolgbarkeit bei Bauteilfehlern innerhalb von 15 Tagen ab 2020
- Umsetzbarkeit mit konventionelle Insellösungen nur innerhalb eines Zulieferers möglich

Zielbild

- Vollständige Zuordnung von Ursprung und Verbauort eines Bauteils
- Rückverfolgbarkeit Herstellerübergreifend gegeben

Lösungsgedanke

- Nutzen der Blockchain-Technologie als konsortiumsöffentliche Datenbank

3.2 Supply Chain & Logistics – JiT / Digital Twin

Ausgangssituation

- Verzögerungen bei Bestellungen entlang der Wertschöpfungskette erzeugen „Bullwhip“-Effekt und führen zu hohen Kosten
- Verzögerungen führen zu Sicherheitsbeständen, Überproduktion oder Lieferausfällen
- Realitätsnahe Abschätzung des Bedarf erzeugt immensen Aufwand

Zielbild

- Beauftragungen und Lieferzeitpunkte werden in die Blockchain geschrieben und können entlang der gesamten Wertschöpfungskette eingesehen werden

Lösungsgedanke

- OEM stellt auf Blockchain-Technologie basierendes System zur Verfügung und gibt darin die Bestellungen ein
- Zulieferer bestätigen Bestellungen und pflegen die Kapazitäten und Lieferzeitpunkte

3.3 Microservices – Computing / Datenökonomie

Ausgangssituation

- IT-Infrastruktur wird zunehmend virtualisiert und bedarfsorientiert belegt
- „Daten sind das neue Öl“
- Derzeit fehlt das Abrechnungsmodell für Kleinstbeträge

Zielbild

- System zur Abrechnung und Buchung von Sub-Cent-Beträgen

Lösungsgedanke

- Automatisierte Buchung und Abrechnung durch die Blockchain-Technologie
- Beispiele: xbr.foundation, dock.io, Augur, Ocean Protocol

3.3 Microservices – Autonomous Mobility / Energy

Ausgangssituation

- In der Elektromobilität gilt das automatische Laden z.B. an der Ampel als Zielbild
- Autonome Mobilität lässt einen neuen Markt an Location Based Services entstehen
- Fehlende Buchungs- und Abrechnungssysteme

Zielbild

- System zur Buchung und Abrechnung einer Vielzahl an Buchungen

Lösungsgedanke

- Automatisierte Buchung und Abrechnung durch die Blockchain-Technologie

3.4 Governance – Verwaltung von Personenstammdaten

Ausgangssituation

- Verwaltung von Personenstammdaten über zahlreiche nationale und internationale Behörden hinweg verursacht immensen Aufwand
- Hohe Schwierigkeiten bei der Festlegung der Datagovernance

Zielbild

- Zugreifbarkeit über Behördenschranken hinweg

Lösungsgedanke

- Alle Beteiligte einigen sich darauf, die Blockchain als konsortiomsöffentliche verteilte Datenbank einzusetzen
- Vorsicht: CRUD-fähig? (EU-DSGVO)

4. Ausblick

What‘s next?

Unverbindliche Prognosen

- Die Situation derzeit ähnelt dem Internet 1990
- Use Cases noch nicht absehbar
- Blockchain-Technologie wird sich als weitere Form der Datenhaltung etablieren
- Anzahl der Kryptowährungen wird stark konsolidiert
- Bedarf an unterschiedlichen Kryptowährungen 2030: max. 15 - 20 Währungen und zahlreiche „Utility Tokens“
- Use Cases durchlaufen einzeln den Gardner Hype Cycle
- Zahlreiche Geschäftsmodelle erst „denkbar“ durch Kryptowährungen und Datenhaltung durch Blockchain
- Weitere Digitale und gesellschaftliche Transformation

5. Q & A

Weiterführende Informationen

Literatur

Antonopoulus, A.: Mastering Bitcoin, O'Reilly, 2. Ed., 2017

Antonopoulus, A.; Wood, G.: Mastering Ethereum, O'Reilly, in print (vorauss. Ende 2018)

Hahn, C.; Wons, A.: Initial Coin Offering (ICO): Unternehmensfinanzierung auf Basis der Blockchain-Technologie, Springer Gabler, 2018

Hofmann, E.; Strewe, U. M.; Bosia, N.: Supply Chain Finance and Blockchain Technology, Springer, 2018

Morabito, V.: Business Innovation Through Blockchain: The B³ Perspective, Springer, 2017

Prusty, N.: Building Blockchain Projects: Building decentralized Blockchain applications with Ethereum and Solidity, Packt Publ., 2017

Sixt, E.: Bitcoins und andere dezentrale Transaktionssysteme: Blockchains als Basis einer Kryptoökonomie, Springer Gabler, 2016

Tanenbaum, A.; van Steen, M.: Verteilte Systeme: Prinzipien und Paradigmen, Pearson, 2. Ed., 2017

Wattenhofer, R.: The Science of the Blockchain, Inverted Forest Pub., 2016

Internet

www.coinmarketcap.com

www.blockexplorer.com

www.bitshares.org

www.xbr.foundation

Abschluss

Haben Sie noch Fragen?

Florian Schütz
florian.schuetz@fom.de

Dr. Peter Vatter
peter.vatter@web.de

FOM Hochschule Nürnberg
www.fom.de/hochschulzentren/nuernberg.html