Kapitel 9: Blockchain und Bitcoin

9: Blockchain und Bitcoin

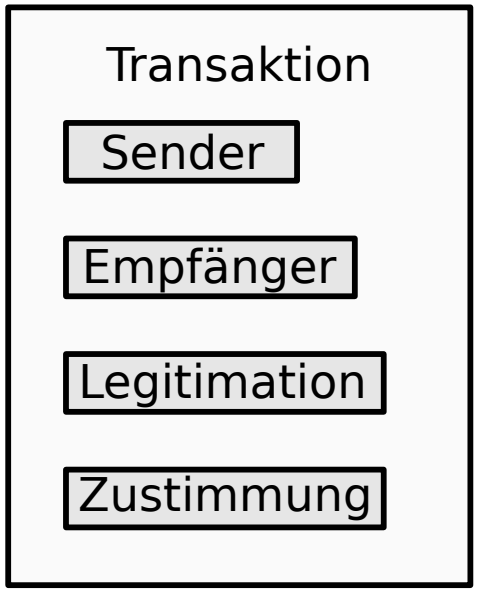
- Bitcoin, (BTC) ist eine kryptographische Währung von Satoshi Nakamoto (Pseudonym)
- Historie
  - 2008: Veröffentlichung des Bitcoin-Zahlungssystem.
  - 2009: Veröffentlichung der Open-Source Software `https://github.com/bitcoin/bitcoin`
  - 2017: 11,5 Millionen Bitcoin-Adressen
- Bitcoin verwendet ein Peer-to-Peer-Netzwerk (Bitcoin-Netzwerk) um Transaktionen zu verwalten.
- Die Transaktionen werden in einer Blockchain gespeichert.
- Es gibt keine zentrale Autorität.

Probleme konventioneller Finanztransaktionssysteme

Bitcoin löst Probleme der konventionellen Finanztransaktionssysteme

- Abhängigkeit von Finanzunternehmen als unabhängige Drittpartei und Buchführer (Bank).
- Vertrauen in die Bank ist notwendig.
- Finanztransaktionen sind prinzipiell reversibel (bei Streitigkeiten kann mit und über die Bank verhandelt werden).
- Ein bisschen Betrug findet sicher statt und ist eigentlich unvermeidlich.
- Dadurch fällt den Banken noch mehr Macht zu.
- Eine Bank kann nur durch Bargeld umgangen werden; online-Zahlungen nur mit Banken möglich.

Überweisung / Transaktionen (Tx)



Welche Informationen werden für eine Überweisung benötigt?

- Sender und Empfänger
- Legitimation des Senders (Sender besitzt das Geld)
- Verifizierbare Zustimmung des Senders

Identitäten

- Eine Identität ist in Bitcoin öffentlichen ECDSA (Elliptic Curve digital Signatur Algorithm) Schlüssel.
- Eine neue Identität wird durch die Generierung eines ECDSA-Schlüsselpaars (**sk**, **pk**) kreiert.
- Eine Bitcoinadresse (Pseudonym, h) ist der Hash des öffentlichen Schlüssels **pk**, d.h. $h = H(\mathbf{pk})$
- Es gilt: $H(x) =$ SHA-256(SHA-256(x))
- Angenommen $\mathbf{pk_A}$ ist Alice öffentlicher ECDSA-Schlüssel, dann ist Alice Bitcoin-Pseudonym $\mathbf{h_A} = \mathbf{H}(\mathbf{pk_A})$.
- Bei Bitcoin kann jeder Nutzer beliebig viele Pseudonyme haben.
- Eine Nutzerin kann durch den Besitz des privaten Schlüssels nachweisen, dass **h** ihr Pseudonym ist. (**Wie?**)

Verifizierbare Zustimmung des Senders

- Der Sender stimmt einer Transaktion zu indem er sie unterschreibt.
- Beim Einlösen von Bitcoins müssen diese mit dem richtigen Schlüssel signiert werden.
- $\mathbf{s} = Sign_{\mathbf{sk}}(Tx)$.
- Eine signierte Transaktion Tx ist ein Tupel $(\mathbf{s}, Tx)$ bei der $\mathbf{s}$ die Signatur von Tx ist.
- Eine signierte Transaktion $(\mathbf{s}, Tx)$ ist valide, falls gilt:

$$\text{Verify}_{\mathbf{pk}}(s, Tx) == \mathbf{true}$$

Legitimität durch vorherige Transaktion

- Legitimation einer Transaktion **Tx** findet durch eine vorher getätigte Transaktion **Tx′** statt.
- Es ist nur möglich Bitcoins zu überweisen, welche man zuvor auch in Form eine Transaktion erhalten hat.
- **Tx** enthält eine Referenz auf **Tx′** welche diese legitimiert.

Problem: Double Spending

Es muss ausgeschlossen werden, dass eine Transaktion Tx_i. mehrfach als Legitimation verwendet wird.

Lösung des Double Spending Problems

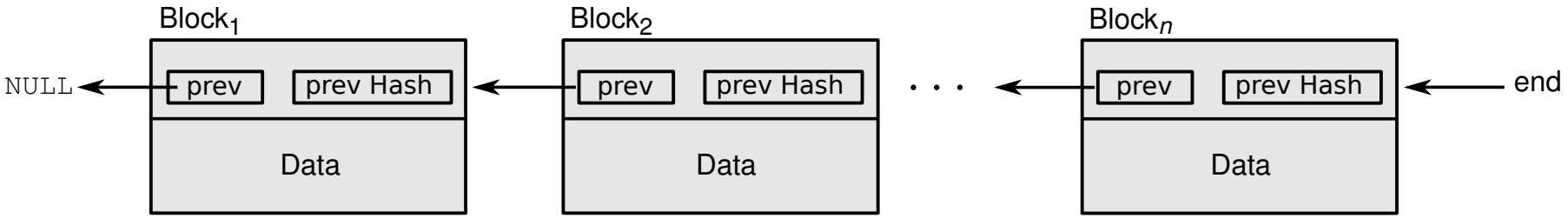
Wie verhindet man Double Spending?

- Nur die erste Transaktion, die Tx_i referenziert, ist gültig.
- $\Longrightarrow$ Eindeutiger zeitlicher Ablauf der Transaktionen wird benötigt.
- Einfach durch eine zentralen Instanz erreichbar.
- Das will Bitcoin aber gerade nicht.
- $\Longrightarrow$ Alle Teilnehmer müssen über alle jemals getätigten Transaktionen Bescheid wissen.
- Lösung: Alle Transaktionen werden in einer, für alle Teilnehmer einsehbaren, Blockchain gespeichert.

9.1: Blockchain

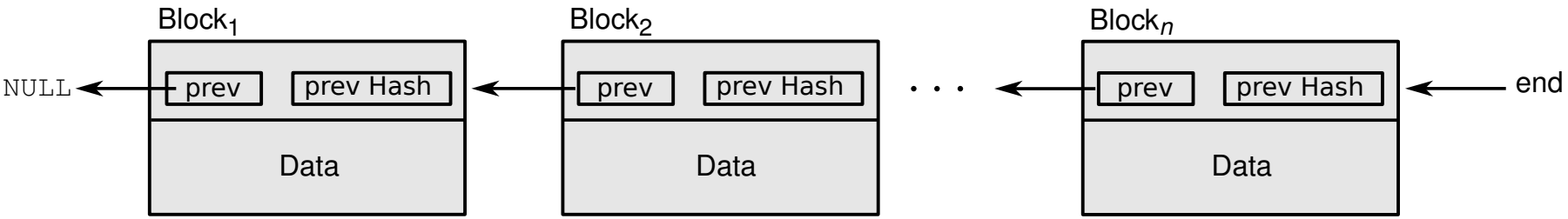
- Die Blockchain ist eine Datenstruktur, welche auf einer kryptographischen Hashfunktion H basiert.
- Mit Hilfe eine Blockchain lassen sich Daten in eine bestimmte Reihenfolge anordnen.
- Jeder kann sich davon Überzeugen, dass die Reihenfolge richtig ist.
- Anwendungsfall: Sicheres (und verteiltes) Logbuch

Die Datenstruktur (1/2)



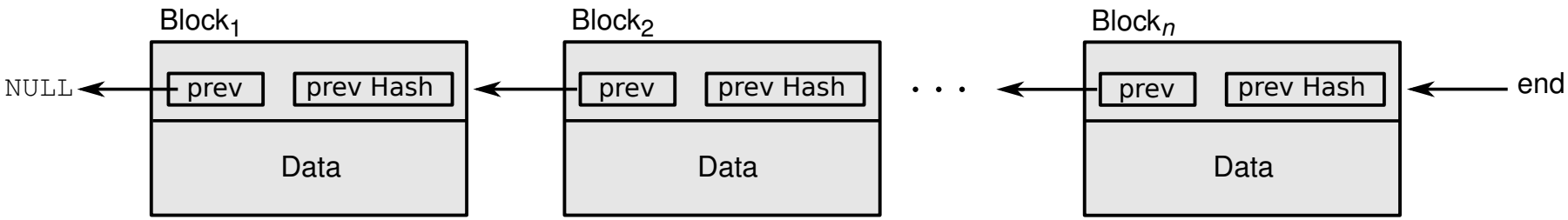
- Die Blockchain ist eine verkettete Liste mit einer Referenz auf den letzten Block/Knoten/Element end und dessen Hashwert H(end).
- Ein Block besteht aus drei Elementen
  - Zeiger auf den vorherigen Block prev
  - Hashwert des vorherigen Blocks prev Hash
  - Gespeicherte Daten Data (Payload)
- Im folgenden sei $\mathbf{B_i}$ der i-te Block.

Die Datenstruktur (2/2)



- Blockhash des i-ten Blocks: $h_i = H(prevHash \,||\, Data_i)$
  Kurzschreibweise: $h_i = H(B_i)$
- Es gilt: $\mathbf{h_1 = H(B_1)}$
- Es gilt: $h_2 = H(B_2) = H(\mathbf{h_1} \,||\, Data_2) = H(\mathbf{H(B_1)} \,||\, Data_2)$
- Es gilt: $h_3 = H(B_3) = H(h_2 \,||\, Data_3) = H(H(B_2) \,||\, Data_3)$
- ...

Gültigkeit einer Blockchain

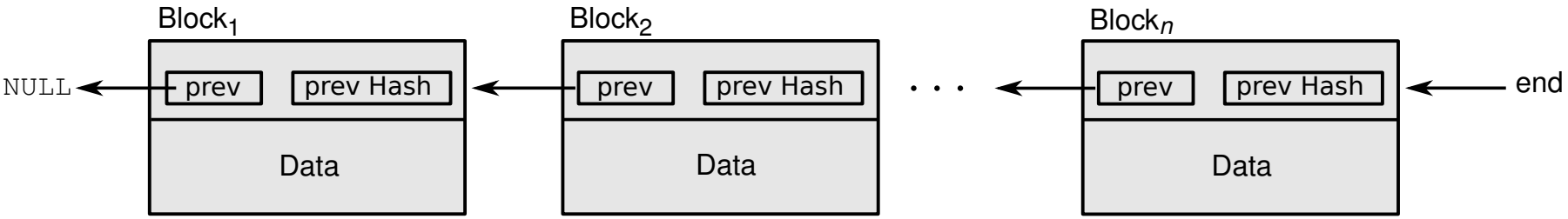


- Sei $prev\_hash_i$ der Wert *prev_hash* des i-ten Blocks.
- Sei $H(\epsilon) = prev\_hash_1$
- Eine Blockchain ist valide, falls
  - für alle $i = 1, \ldots, n$ gilt:

$$\mathbf{H(B_{i-1}) = prev\_hash_i}$$

  - Es gilt: $\mathbf{H(end) = H(B_n)}$
- Die Blockchain ist also valide, wenn alle Hashwerte stimmen.

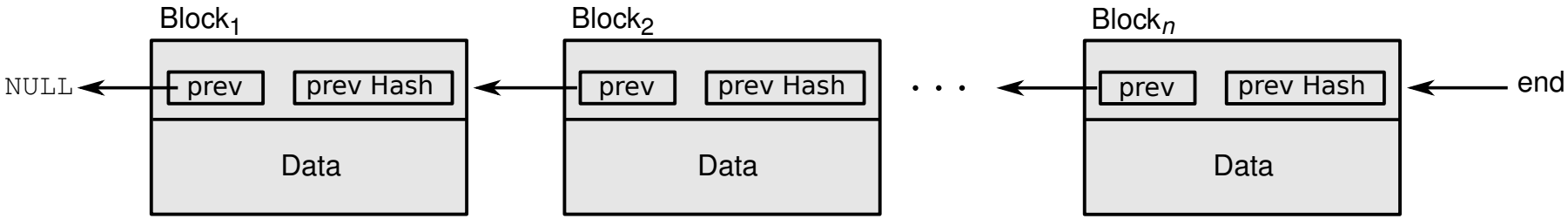
Sicherheit (1/2)



- Das Ergebnis einer erfolgreichen Manipulation einer validen Blockchain **X** ist eine neue valide Blockchain **X′**.

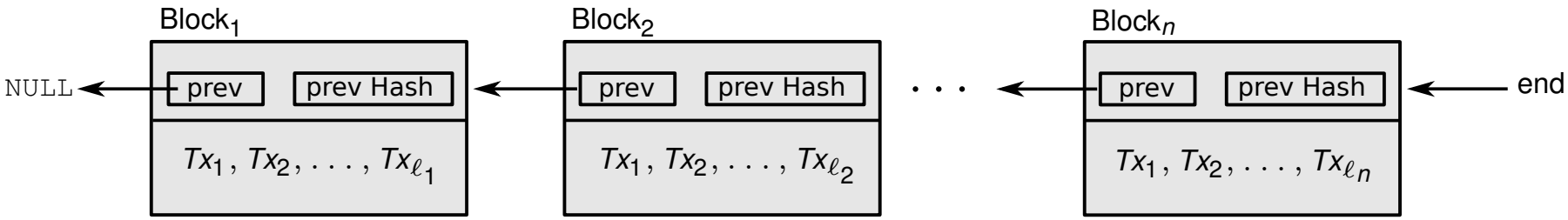
- **Frage:** Wie einfach/schwierig ist eine solche erfolgreiche Manipulation?

Sicherheit (2/2)



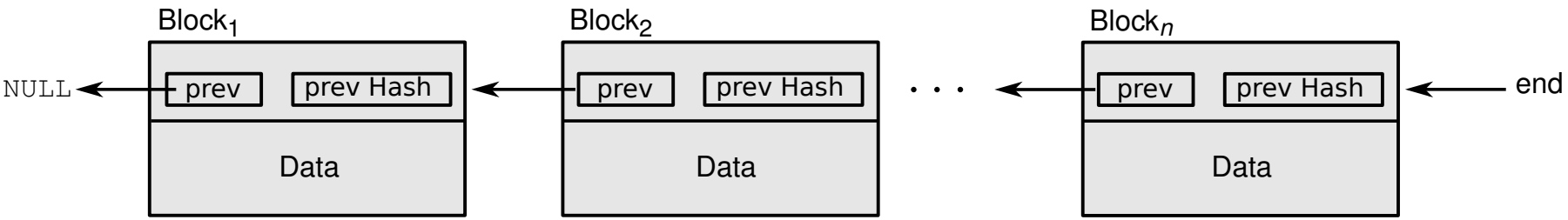
- Beobachtung: Für eine erfolgreiche Manipulation muss ein Angreifer eine Kollision für einen Block finden.
  - i) Im Voraus $\mathbf{B_i}$ und $\mathbf{B_i'}$ mit $H(\mathbf{B_i}) = H(\mathbf{B_i'})$ generieren.
  - ii) Im Nachhinein einen Block $\mathbf{B_i'}$ finden für den gilt: $H(\mathbf{B_i'}) = \mathbf{h_i}$.
- In beiden Fällen kann $\mathbf{B_i}$ durch $\mathbf{B_i'}$ ausgetauscht werden, um eine erfolgreiche Manipulation durchzuführen.
- **Frage** Was ist der Aufwand für i) bzw. ii)?

Legitimität durch Blockchain



- Legitimation einer Transaktion **Tx** findet durch eine vorher getätigte Transaktion **Tx′** statt.
- Es ist nur möglich Bitcoins zu überweisen, welche man zuvor auch in Form eine Transaktion erhalten hat.
- **Tx** enthält eine Referenz auf **Tx′** welche diese legitimiert.

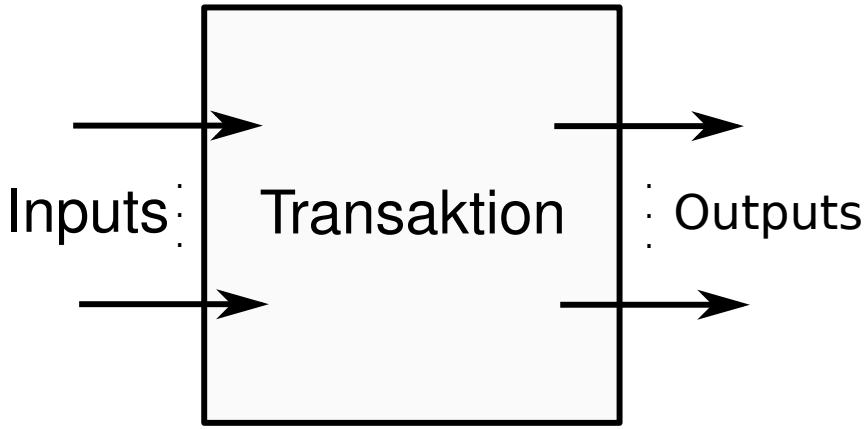
Bitcoin: Doublespending



- **Frage:** Was muss ein Angreifer tun, um bei Bitcoin Double Spending zu betreiben?

- **Frage:** Wie hoch ist der Aufwand dafür?

9.2: Bitcoin Transaktion



- Inputs
  - Referenz auf Transaktion **Tx′**
  - Signatur **s** von **Tx′**
  - Öffentlicher Schlüssel des Senders **pk**
- Outputs
  - Betrag
  - Empfängeradresse (**id** = H(**pk**))

Mehrere Inputs und Outputs

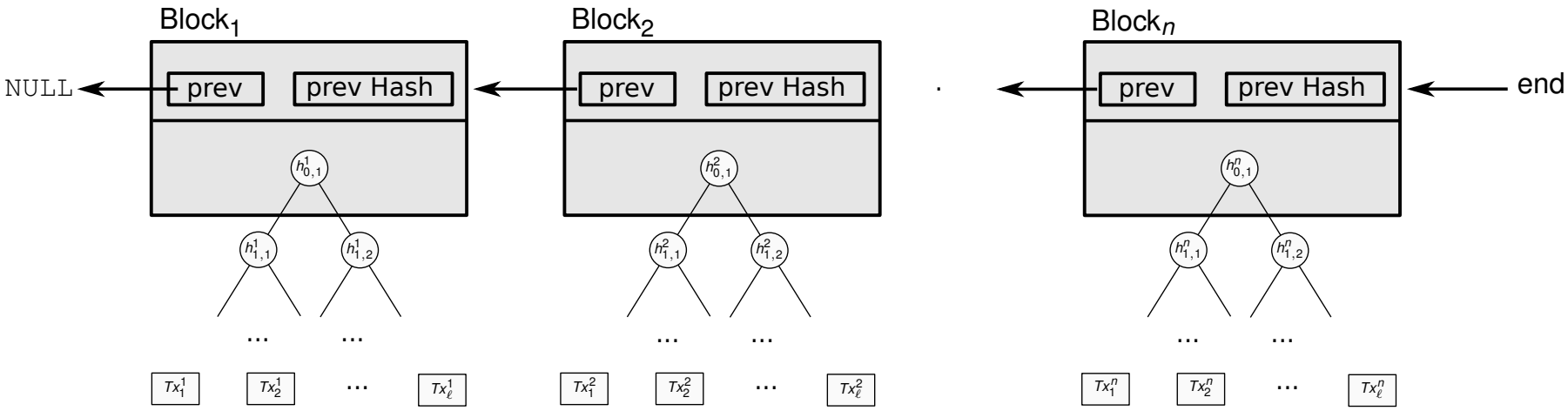
Folgen der Bitcoin-Legitimation

- Alice kann nur Beträge (Bitcoins) überweisen, welche sie auch erhalten hat.
- Angenommen Alice Bitcoinadresse ist der Empfänger einer Transaktion über **Tx** 0,1 Bitcoins.
- **Frage:** Wie kauft Alice von Bob eine Pizza für 0,001 Bitcoin?
- Alice erstellt eine Transaktion mit zwei Outputs.
  - Bobs Bitcoinadresse (0,001 Bitcoin).
  - Eine Bitcoinadresse von Alice (0,099 Bitcoin).

Validierung von Transaktionen

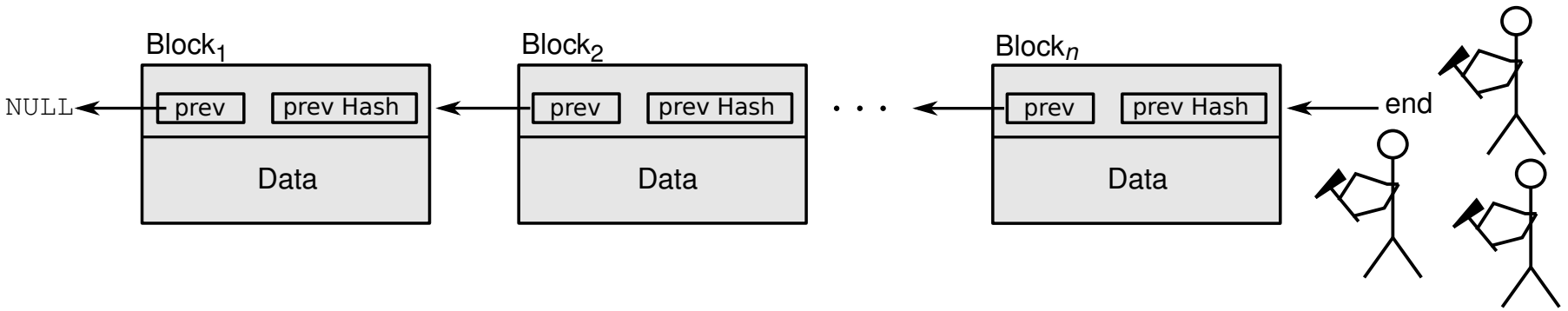
- Die Summe der Input-Beträge = Summe der Output-Beträge
  Ausnahme: Mining-Transaktion
- Für alle Inputs wird Folgendes getestet
  1. Transaktion **Tx′** ist frisch und valide
  2. **Tx.outputs.id′** = **H**(**pk**) existiert
  3. $\text{Verify}_{\mathbf{pk}}(\mathbf{s}, \mathbf{Tx'}) == \mathbf{true}$
- Beobachtungen
  - Empfängeradresse wird nicht verifiziert.
  - Damit können Bitcoins nach `/dev/null` gesendet werden.
  - Signatur **s** und öffentlicher Schlüssel **pk** muss erst beim Einlösen von Bitcoins (Inputs) preisgegeben werden.
  - Damit schützt man sich vor ECDSA-Angreifern.

Bitcoin Block



- Die Blockgröße wurde 2010 auf 1MB limitiert.
- Eine durchschnittliche Transaktion benötigt 250 Bytes.
- In einen Block passen ca. 2500 Transaktionen.
- Die Transaktionen eines Blocks werden mit Hilfe eines Merkle-Trees verwaltet.

9.3: Bitcoin Mining



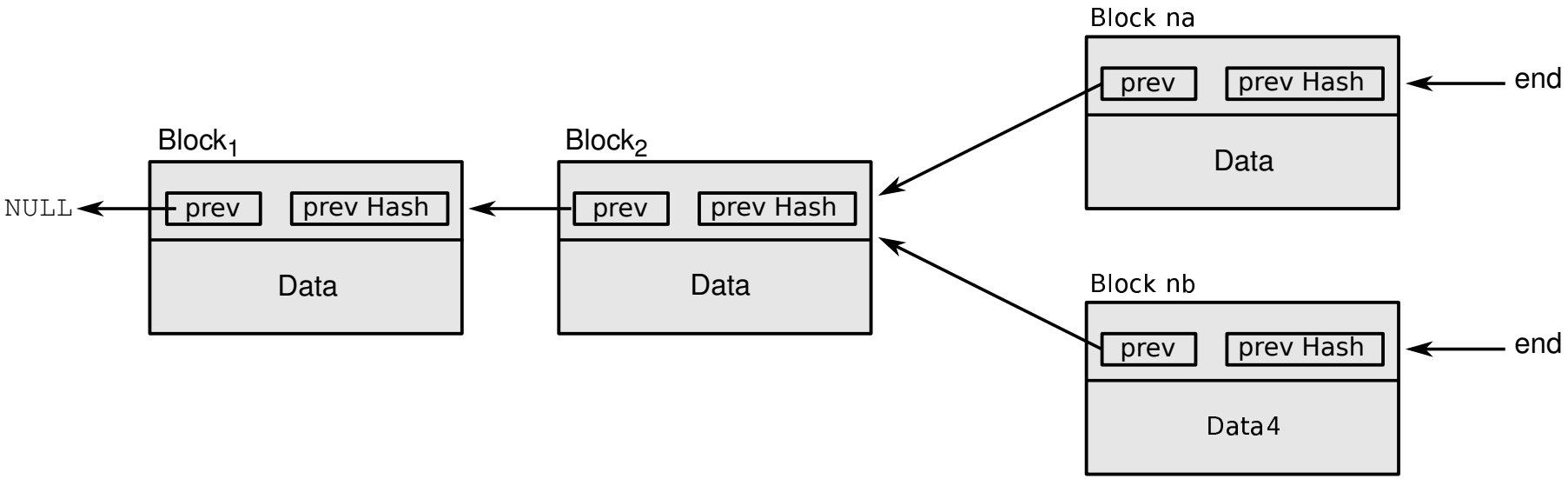
Mining

Mining bezeichnet das Erstellen und Anhängen eines neuen validen Blocks an die Bitcoin-Blockchain.

Block Mining Reward

- Miner sammelt zunächst frische und valide Transaktionen.
- Sind genügend beisammen, beginnt der Miner mit dem Mining.
- Für das Minen bekommt der Miner Bitcoins (Block Mining Reward).
  - Die Belohnung pro generierten Block war zu Beginn 50 BTC.
  - Alle 210.000 Blöcke (ca. 4 Jahre) wird die Belohnung halbiert.
  - Im Schnitt wird alle 10 Minuten ein neuer Block generiert, d.h. pro Jahr sind dies ca. 67.000.
- Block Mining Reward: Transaktion ohne Input und festem Output-Betrag von derzeit 12.5 BTC.
- Neue Bitcoins können ausschließlich durch Minen generiert werden.
- Jeder Bitcoin-Client kann minen.

Blockchain Fork



- Was passiert falls zwei Miner parallel einen neuen Block minen?
- Es darf immer nur einen Nachfolger geben, wer bekommt also recht?
- In dem Fall führen alle Knoten zunächst zwei Blockchains parallel fort (Fork).

Consensus

- Angenommen es gab bei der Bitcoin-Blockchain BBC einen Fork.
- Jeder Miner entscheidet für sich, für welche der beiden BBCs er minen möchte.
- Sobald eine BBC länger ist als Ihr Fork wird dieser verworfen.
- Consens: Die längste BBC gewinnt, d.h. sobald eine längste Kette feststeht, werden alle anderen verworfen.

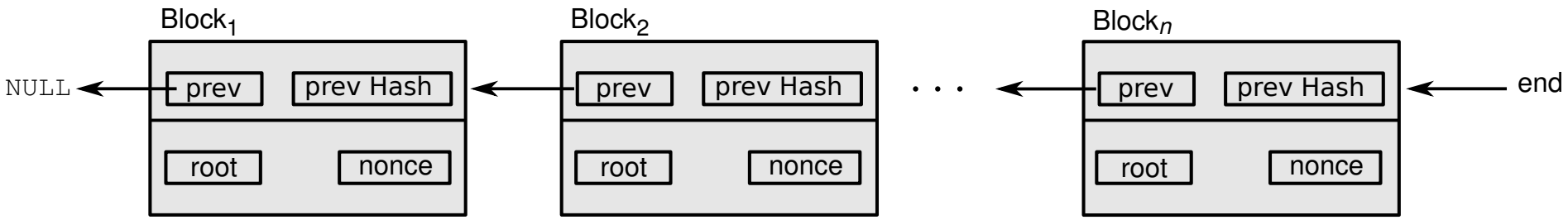
Dominierungs Problem

- Noch ein Problem: Da immer die längste BBC gewinnt, könnte ein Miner dominieren, weil er schneller rechnen kann und eine bessere Netzverbindung hat.
- Ein solcher Miner ist besser als alle andere.
- Damit könnte er schneller Transaktionen sammeln und schneller Blöcke veröffentlichen.
- Ein solcher Knoten wäre de facto die Bank; dies ist unerwünscht.
- Es wird ein Mechanismus benötigt, der dafür sorgt, dass ein einzelner Miner nur dominieren kann, wenn er mehr Rechenkraft aufbringt als der Rest des Netzwerks zusammen.
- Lösung: Proof of Work

Proof of Work

- Bei Bitcoin wird verlangt, dass der die Binärdarstellung des Hashwerts eines jeden Blocks eine bestimmten Präfix (z.B. Anzahl führende Nullen) besitzen muss.
- Je länger der Präfix, desto höher der Schwierigkeitsgrad.
- Der Schwierigkeitsgrad wird alle 2016 Blöcke angepasst.
- Ziel der Anpassung: Miningdauer soll 10 Minuten sein.
- Der Schwierigkeitsgrad passt sich also dynamisch der Rechenleistung der Miner an.
- Um das Ziel zu erfüllen wurde der Bitcoin-Block um eine Nonce erweitert, welche der Miner frei wählen kann.

Mining Pseudocode



Mining(prefix, BBC):

1: $Tx \leftarrow freshTransactions()$;
2: $B \leftarrow newBlock(Tx)$;
3: $B.nonce \leftarrow random()$;
4: **while** $H(B).startsWith(prefix) =$ **false do**
5: B.updateNonce(); {Increment nonce}
6: **end while**
7: BBC.append(B);
8: **return** BBC;

Bitcoin Mining

Sei **p** die Länge des geforderten Bitcoin-Präfixes **P**, dann gilt

$$Pr\left[H(B).startsWith(\mathbf{P})\right] = \frac{1}{\mathbf{2^p}}.$$

- Der Mining-Prozess kommt also einem Glücksspiel gleich.
- Wer mit einem 2^p seitigen Würfel zuerst **P** würfelt hat gewonnen.

- Schnelles Würfeln erhöht die Chance als erstes einen neuen validen Block gefunden zu haben.

- **Frage:** Wie könnte ein dominierender Angreifer *betrügen*, der pro Sekunde mehr würfelt als alle anderen Teilnehmer zusammen?

Transaktionsgebühren

- Bei Bitcoin gibt es Transaktionsgebühren.
- Bei Bitcoin ist es erlaubt, das der Input-Betrag einer Transaktion größer als dessen Output-Betrag ist.
- Die Differenz muss der Miner an eine beliebige Bitcoinadresse überweisen.
- **Frage:** Was passiert, falls der Miner dies nicht macht?
- **Frage:** Warum zahlen Nutzer überhaupt Transaktionsgebühren?
- **Frage:** Welche Kriterien würdet Ihr als Miner für den nächsten Block auswählen?

Bitcoin Genesis Block

```
01000000 - version
...0000000000000000000000000000000000000000000000000000000000000000 - prev block
...B12B27AC72C3E67768F617FC81BC3888A51323A9FB8AA4B1E5E4A - merkle root
29AB5F49 - timestamp
FFFF001D - bits
1DAC2B7C - nonce
01 - number of transactions
01000000 - version
01 - input
...0000000000000000000000000000000000000000000000000FFFFFFFF - prev output
4D - script length
...6F66207365636F6E64206261696C6F757420666F722062616E6B73 - scriptsig
FFFFFFFF - sequence
01 - outputs
00F2052A01000000 - 50 BTC
43 - pk_script length
...CEF38C4F35504E51EC112DE5C384DF7BA0B8D578A4C702B6BF11D5FAC - pk_script
00000000 - lock time
```

9.4: Zahlen und Fakten

Stand Februar 2023

- Länge der Blockchain: 510.227 Blöcke
- Größe der Blockchain: 452,95 GB
- Aktive Bitcoinadressen: ca. 688.000
- Anzahl BTC: ca. 29,3 Millionen
- Kurs: ca. 23.254 USD
- Gesamtwert: ca. 681,3 Milliarden USD
- Durchschnittliche Transaktionsgebühren: ca. 1,69 USD

Bitoin Minning Epochen

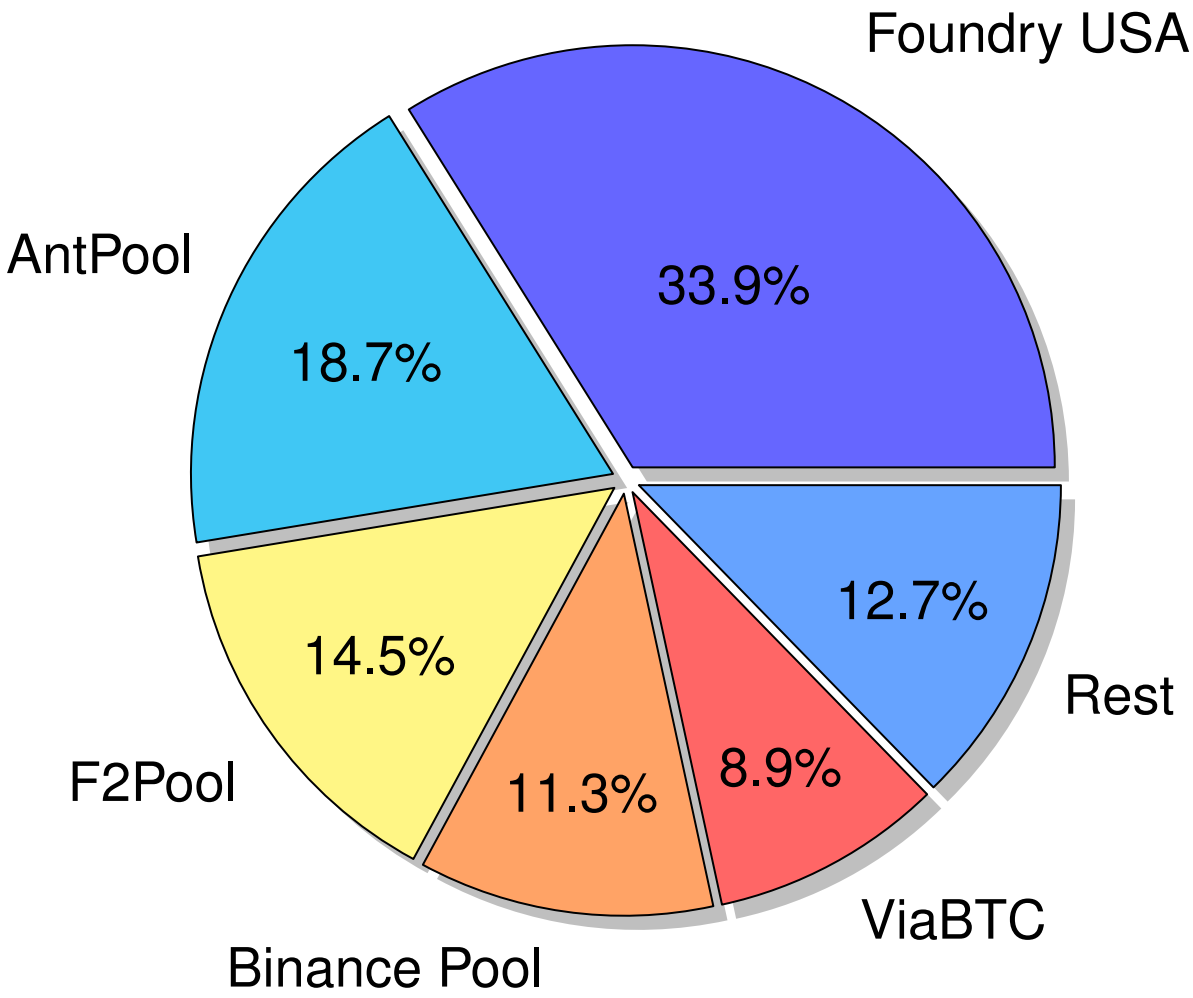
- bis 10.2010: CPU
- seit 11.2010: Mining Pools entstehen
- bis 06.2011: GPU
- bis 01.2013: FPGA
- seit 2013: ASIC

Bitcoin Mining Farm

Quelle: `http://www.datacenterknowledge.com`, KnC Miner

Bitcoin Mining Farm KnCMiner, Schweden.

Marktanteile der Top 5 Mining Pools (Februar 2023)



(https://blockchain.info/pools?timespan=4days)

Kritik: Hohe Energiekosten

- Proof of Work = Proof of Power Consumption
- Alternative: Proof of Storage
- Zahlen vom 21. Februar 2023

  (`https://digiconomist.net/bitcoin-energy-consumption`)

  - Jährlicher Strombedarf von Chile
  - Stromverbrauch (Tag): 218,871 GWh
  - Versorgbare Zwei-Personen-Haushalte: 34,74 Millionen
  - Stromverbrauch (Transaktion): 728,41 KWh
  - CO2-Verbrauch pro Transaktion: 406,5 g

Weitere Informationen

- http://learnmeabitcoin.com
- https://en.bitcoin.it/wiki/Main_Page
- https://blockchain.info
- https://github.com/bitcoin/bitcoin
- https://www.bitcoin.de/

9.5: Bonuskapitel: Smart Contracts

- Nutzen einer Blockchain (zentralisiert oder dezentralisiert)
- Anstatt, wie im Bitcoin-Netzwerk, nur Transaktionen zu sammeln,
- Sammelt man zusätzlich dazu Algorithmen (Skripte) in der Blockchain.

Beispiel

A versichert sich bei B gegen einen Wahlerfolg von Donald Trump

```
A = a890289023f44ß0ad9
B = 9ddd74905c94f89380

at "UTC 2017-01-21-23-59":
    N = query(who-is-us-president/trustworthy.org)
    N = to_upper(N)
    if N == "CLINTON":
       transfer 100 from A to B
    else:
       transfer 500 from B to A
```

+ keine Rechtsanwälte, keine Richter
− keine Möglichkeit, Fehler oder Missverständnisse zu korrigieren (wenn, z.b. trustworthy.org mit “HCLINTON” antwortet)

Smart Contracts – kann da noch etwas schiefgehen?

A versichert sich bei B gegen starke Kursschwankungen
2017-05-29: 1 EUR = 1.1166 $

```
A = a890289023f44ß0ad9
B = 9ddd74905c94f89380

def f(n):
  if n < 3: return 1
  else: return f(n-f(n-1)) + f(n-f(n-2))

transfer 10 from B to A
at "UTC 2018-05-29-12-00":
    Rate  = query(euro-per-us-dollar/exchange-rates.de)
    Money = f(round(abs((1/Rate-1.1166)*100)))
    transfer Money from A to B
```

Zusammenfassung

- Sie sollten wissen, was eine Blockchain ist und was man mit ihr machen kann.
- Sie sollten wissen, wann eine Blockchain sicher ist.
- Sie sollten das Double Spending Probleme verstanden haben.
- Sie sollten die Stärken und Schwächen von Bitcoin kennen.
- Sie sollten wissen, wie Bitcoins grundsätzlich funktionieren.