Wie funktionieren Bitcoin?

Sharru Moeller

März 26, 2018

Hintergrund: Kryptowährungen

- Digitales Zahlungsmittel
- Dezentralisiert
- Nutzt kryptographische Verfahren
- Beispiele:
  - Bitcoin
  - Ethereum
  - Litecoin
  - ...

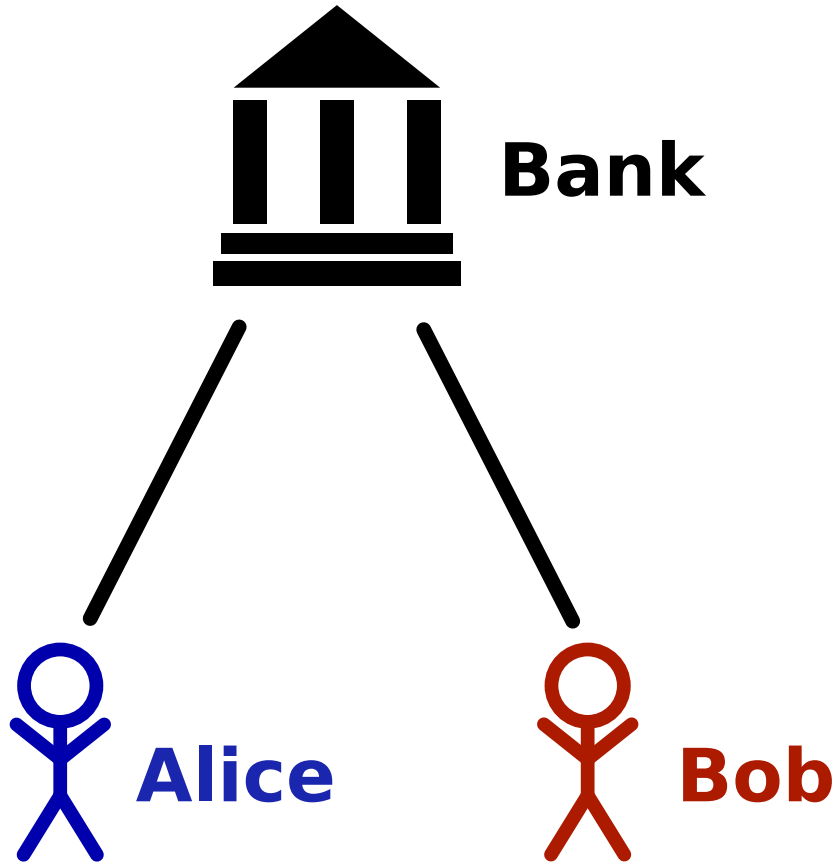
Bitcoin Logo

Grundidee Kryptowährungen

Transaktionshistorie

Alice	zahlt ein	300FR
Alice	zahlt **Bob**	200FR
Bob	zahlt **Alice**	100FR

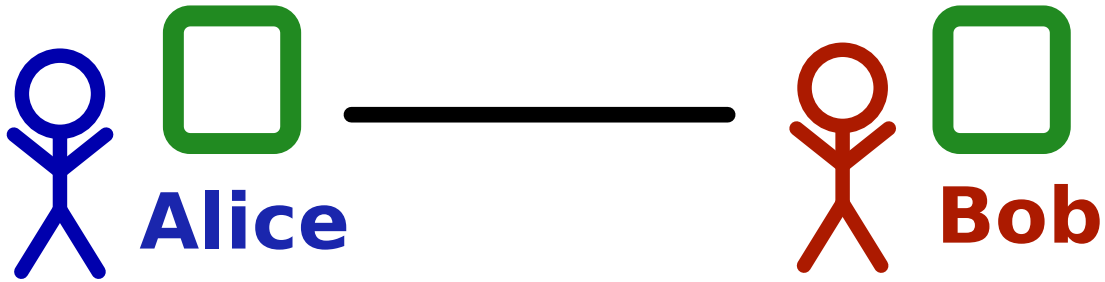
Alice	200FR
Bob	100FR



Dezentralisierung

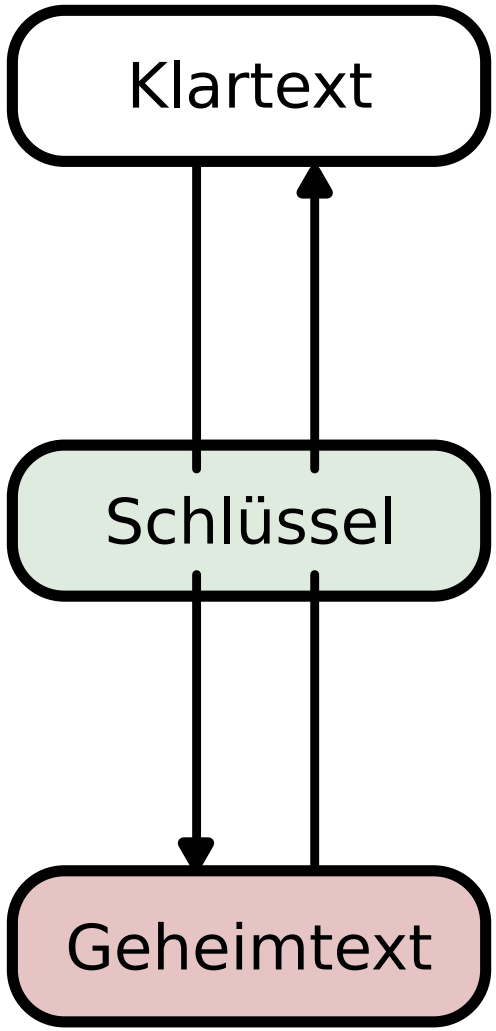
- Jeder besitzt Kopie der Transaktionshistorie
  $\Rightarrow$ Öffentlich Einsehbar
- Selbstregulierend
- Keine zentrale Instanz

$\Rightarrow$ **Probleme**



Symmetrische Verschlüsselung

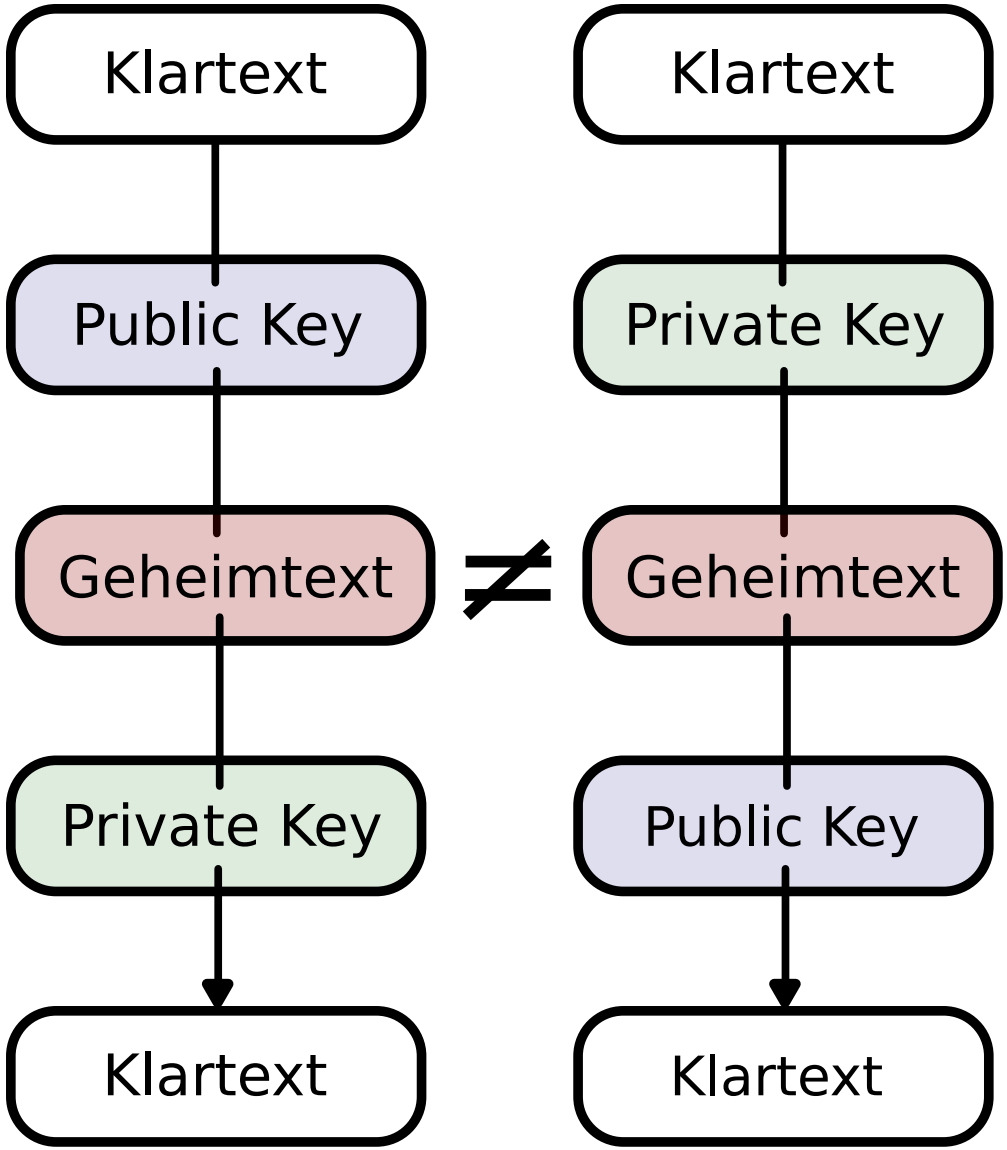
- Ein Schlüssel
  $\Rightarrow$ Ver- und Entschlüsselung
- Anwendung:
  z.B. Internet Verbindung



Exkurs: Kryptographie

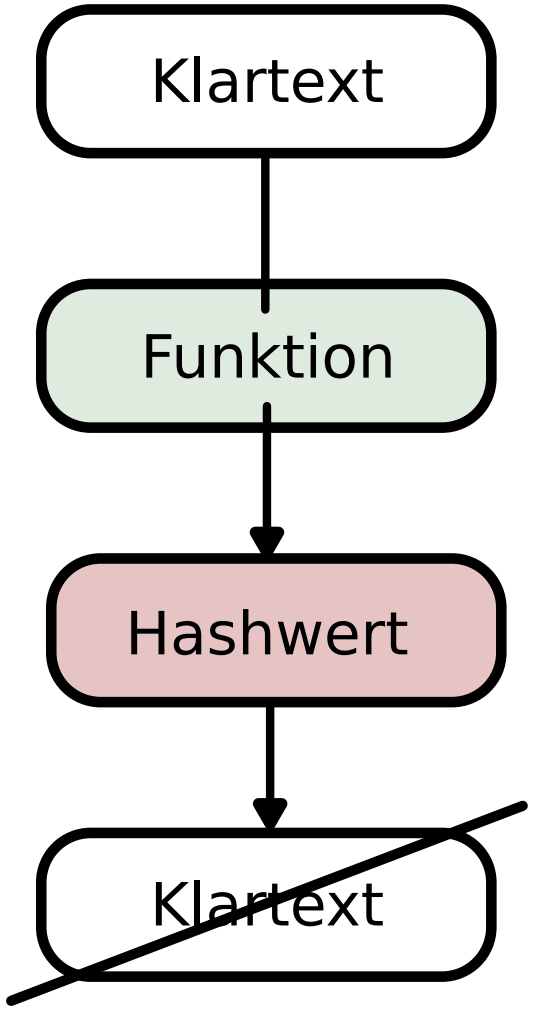
Asymmetrische Verschlüsselung

- Zwei Schlüssel (Public und Private)
- Private Key nicht durch Public Key ableitbar
- Anwendung: z.B. Zertifizierung



Hash Verfahren

- Hashwert generiert aus Klartext
- Feste Länge
- Wiederherstellung des Klartextes unmöglich
- Beispiel: "Hallo" $\Rightarrow$ a4cf
- Anwendung:
  z.B. Passwortabgleich



Problemstellung

Dezentralisierung

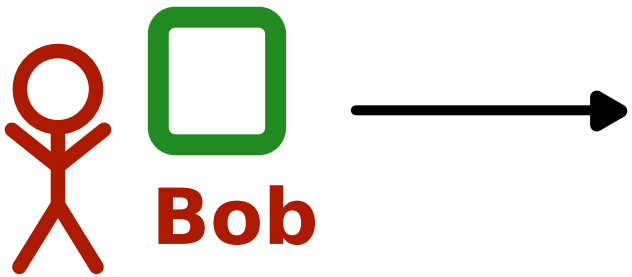
- Jeder besitzt Kopie der Transaktionshistorie
  $\Rightarrow$ Öffentlich Einsehbar
- Selbstregulierend
- Keine zentrale Instanz

$\Rightarrow$ **Probleme**

Autorisierung

Problem: Unautorisierte Transaktionen

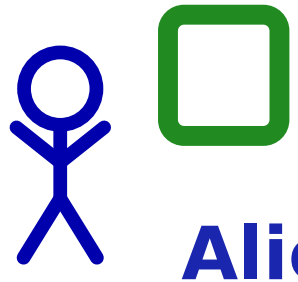
$\Rightarrow$ Empfänger/Dritter schreibt Transaktionen



Transaktion

Alice zahlt **Bob** 100FR

Autorisierung



Alice

Public Key **A1**

Private Key **A2**

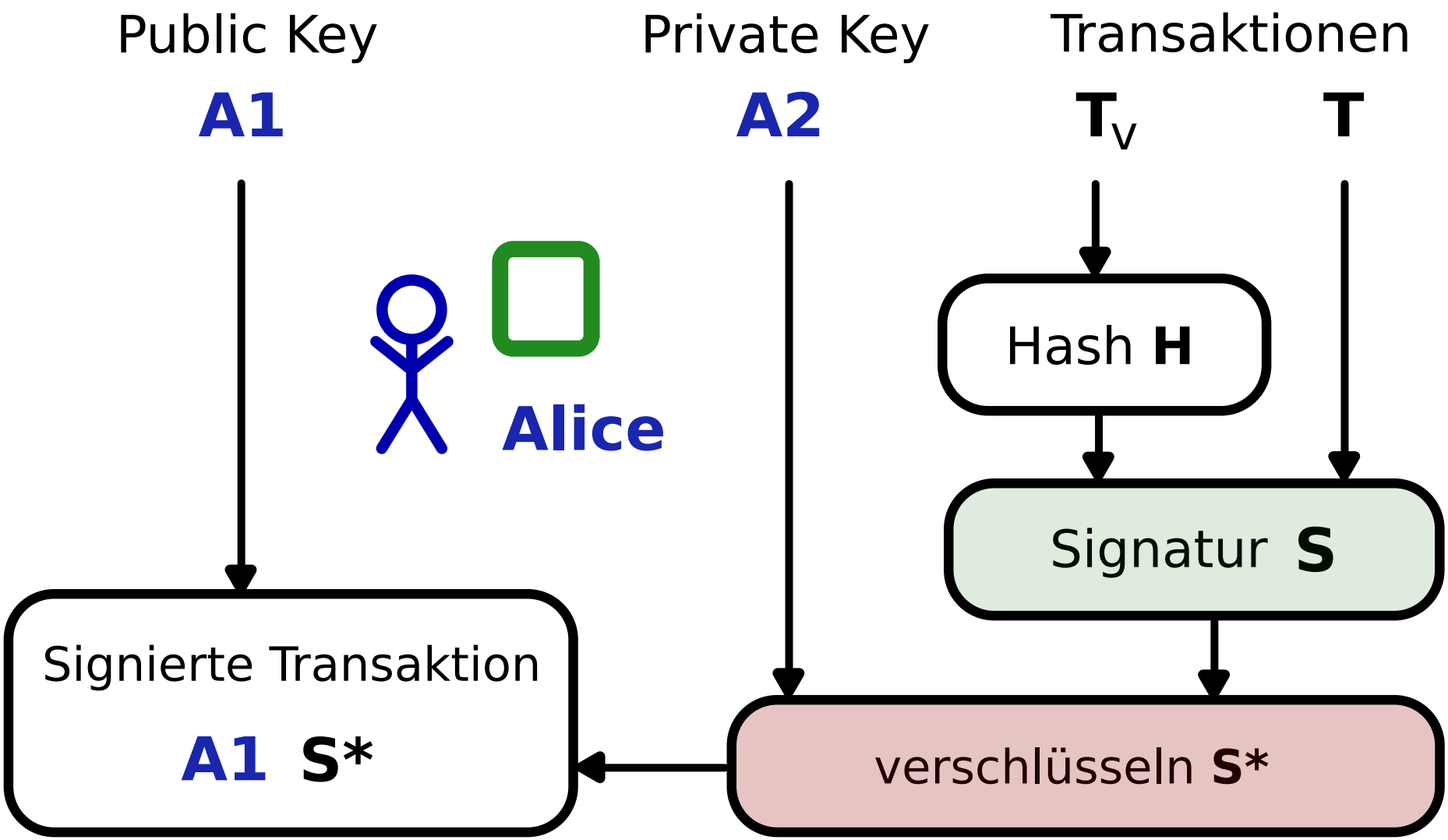
Signatur **S**

Identifikation

- Akteure durch Public Key identifiziert
- Signatur verschlüsselt durch Private Key
  $\Rightarrow$ Eindeutige Identifikation
- Beispiel:

  A1 zahlt **B1** 100FR
  $\Rightarrow$ **A1** **S***

Autorisierung



Signieren

- Signierung durch asymmetrische Verschlüsselung
- Absender durch Public Key eindeutig
- Signatur basiert auf vorige Transaktion

Transaktion: **A1** **S***

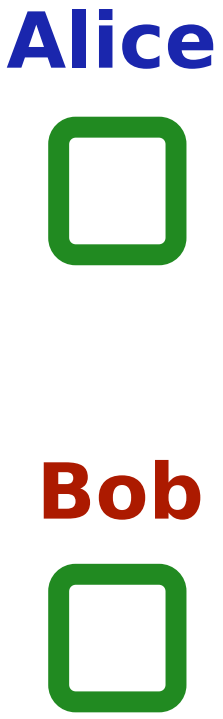
Blockchain: Idee

Probleme

- Welche Transaktionshistorie?
- Verändern alter Transaktionen

Idee

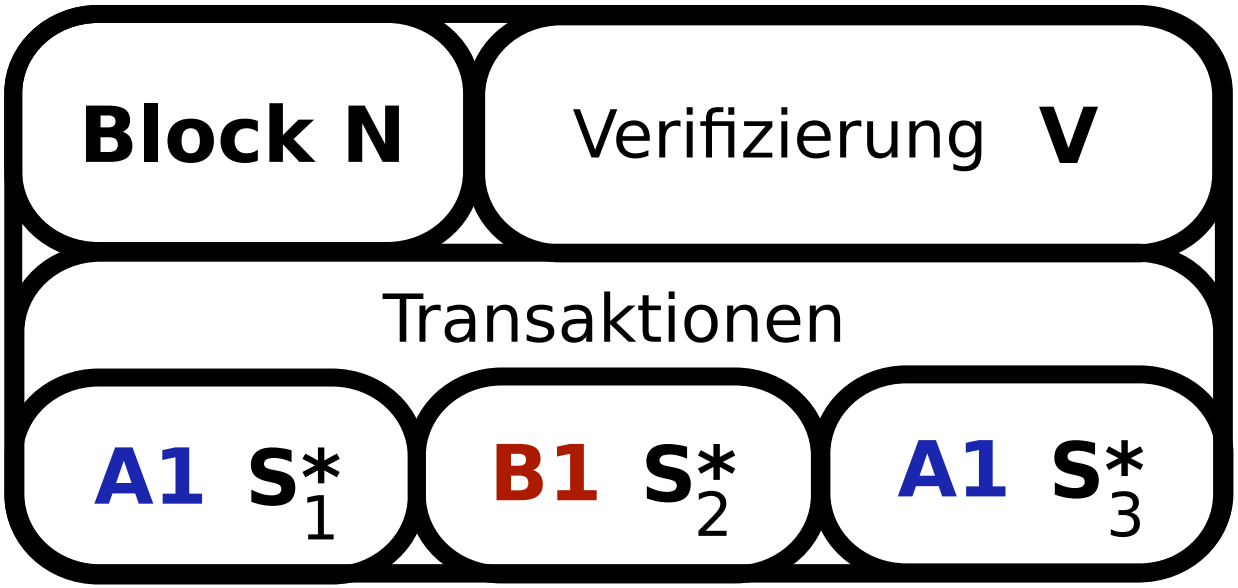
- Bilden einer Kette
- Verändern eines Ketteglieds zerstört Kette
- Längste Kette als korrekt akzeptiert



Blockchain: Bildung

Block

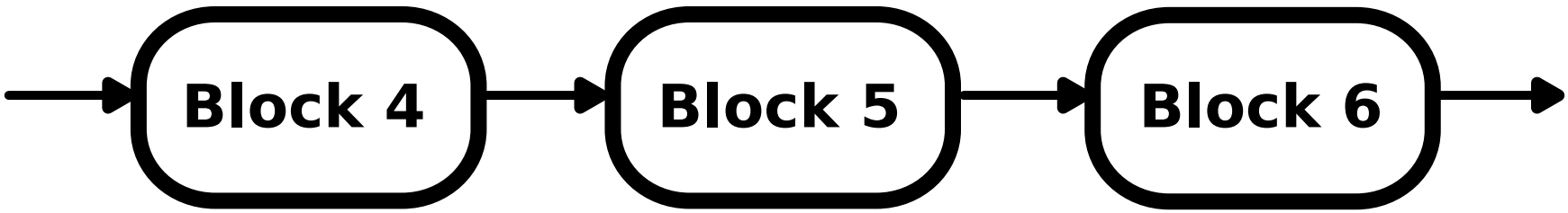
- Sammlung an Transaktionen
- Block verifiziert durch "Proof-of-Work"



Blockchain: Bildung

Blockchain

- Verkettung von Blöcken durch Abhängigkeiten
- Bildet die gesamte Transaktionshistorie



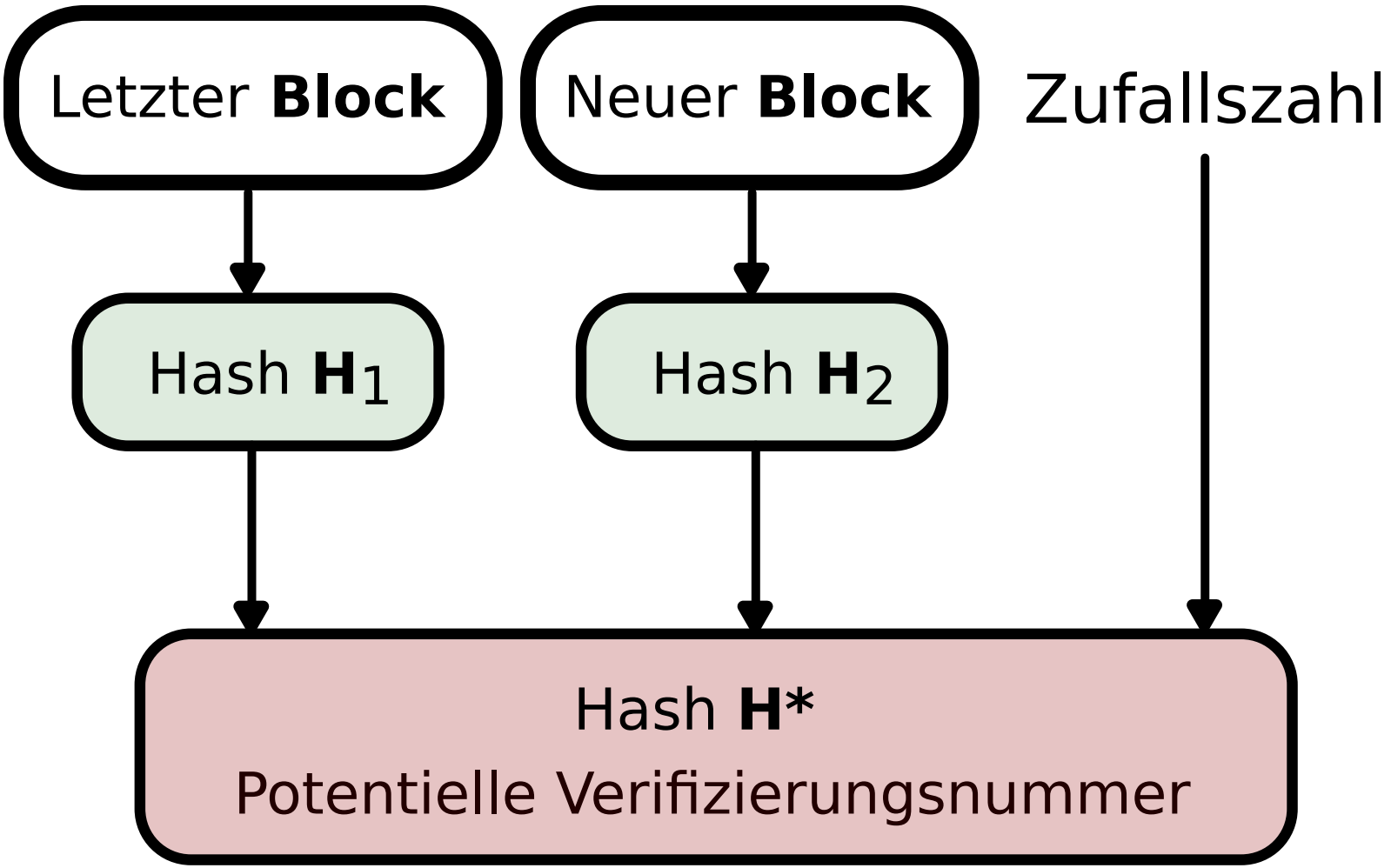
Blockchain: Verifizierung

Proof of Work

- Basiert auf einer Hashfunktion
- Erfolgreicher Fund bei N beginnenden Nullen
- Teilnehmer testen Zufallszahlen
- Beispiel:

  "Hello, world!0" $\Rightarrow$ 1312af178c253f8..
  "Hello, world!1" $\Rightarrow$ e9afc424b79e4f6..
  ...
  "Hello, world!4250" $\Rightarrow$ **0000**c3af42fc311...

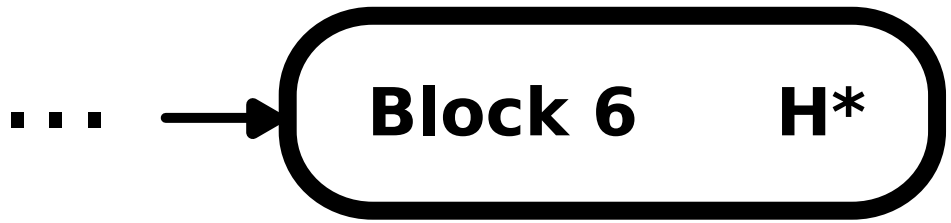
Blockchain: Verifizierung



Blockchain: Verifizierung

Verifizierung von Blöcken

- Block erhält Verifizierungsnummer
- Referenz auf den vorigen Block
- Längste Kette benötigte meiste Arbeit
  $\Rightarrow$ Längste Kette als Transaktionshistorie



Blockchain: Verifizierung

Wer verifiziert?

- Einteilung in "Miner" und "Nutzer"
- "Miner" auch als Knoten bezeichnet
- Knoten arbeiten und verifizieren
- Entlohnung in Bitcoins
- Knoten benötigen volle Historie

Nutzer

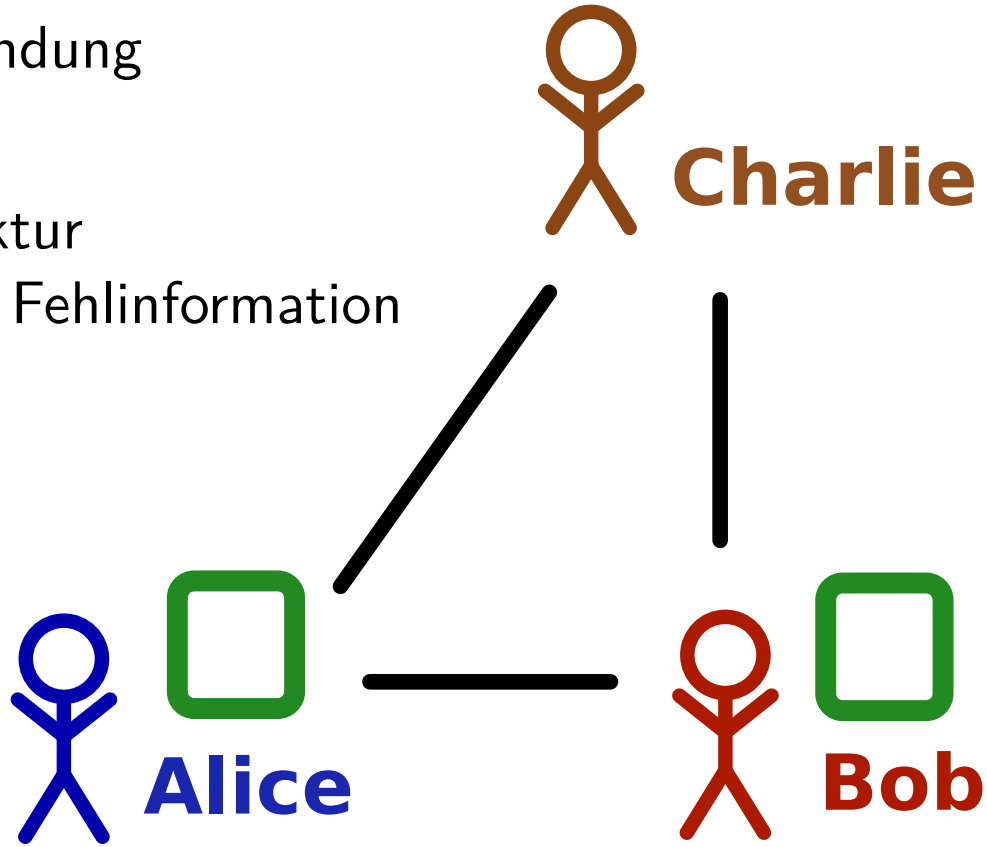
Nutzt Bitcoin

Miner

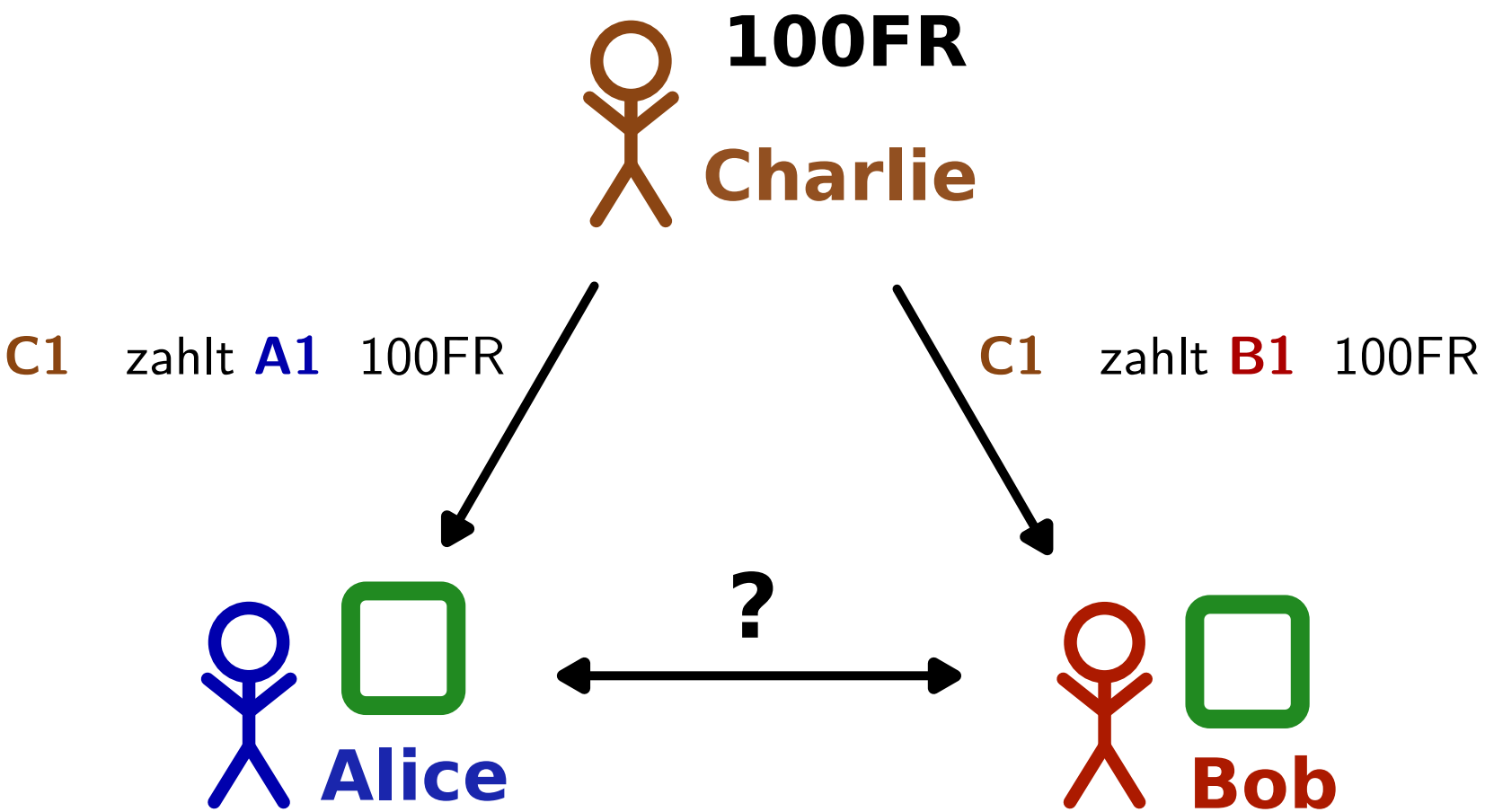
Suchen **H*** für neuen Block

Problem: Double Spending

- Beschreibt mehrfache Verwendung desselben Geldes
- Möglich durch virtuelle Struktur z.B zeitliche Ungenauigkeit, Fehlinformation

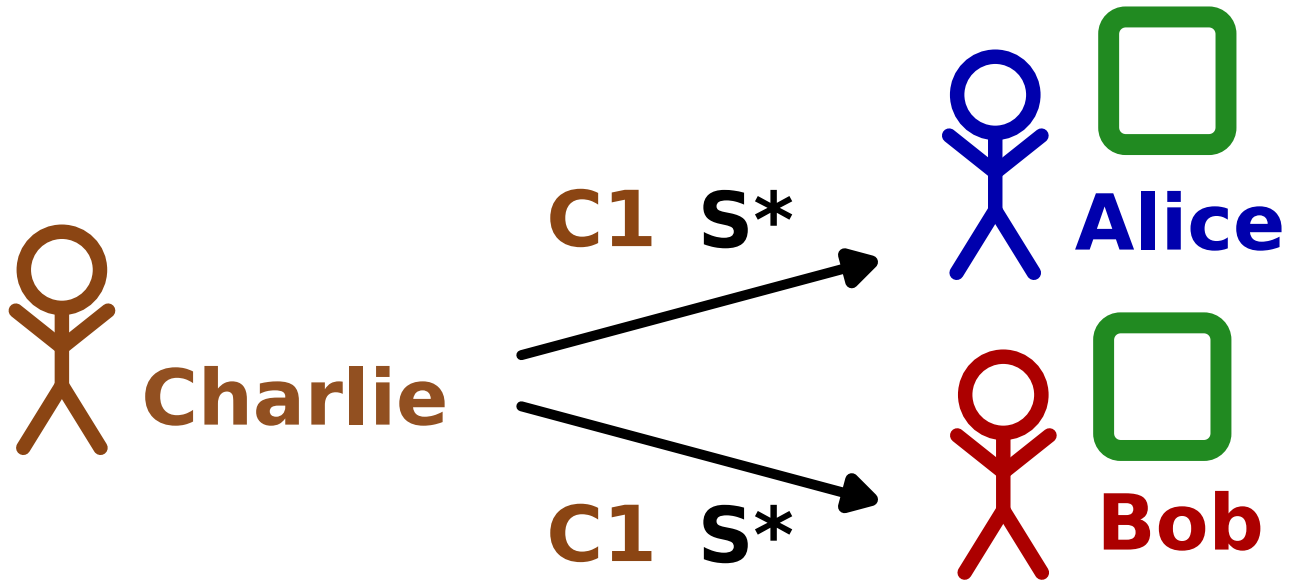


Double Spending



Verhinderung von Double Spending

- Transaktionen an alle Teilnehmer senden
- Kontostand immer bekannt durch Historie
- Anfügen eines Zeitstempels an Blöcken



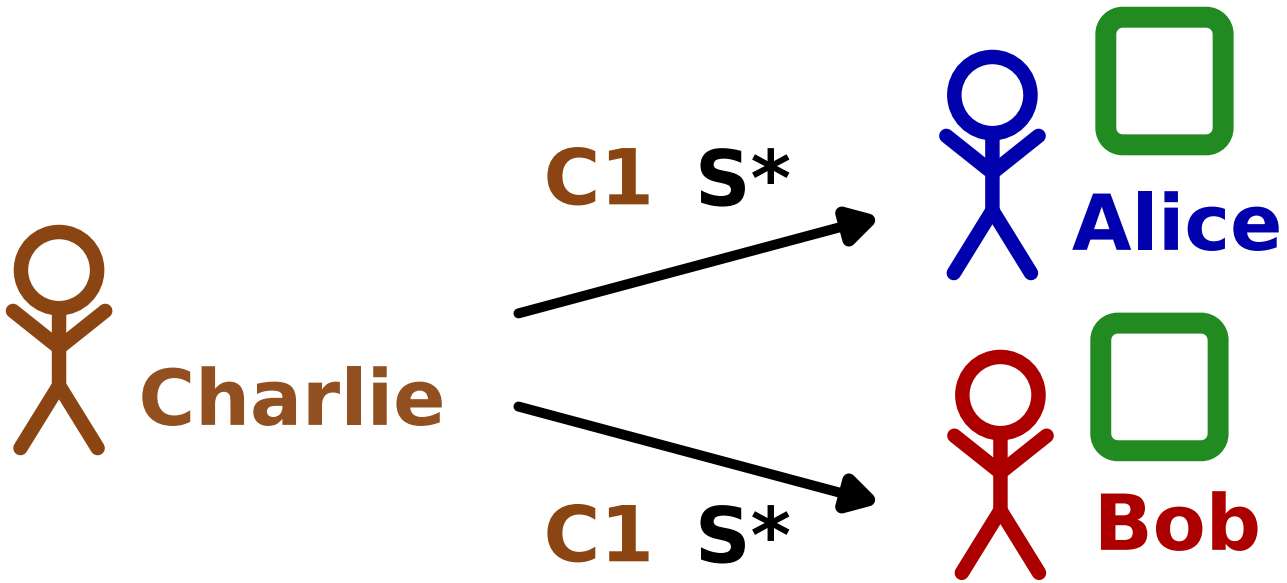
Zusammenfassung: Protokoll

Überblick Protokoll

1. Neue Transaktionen an alle Knoten senden
2. Sammeln von Transaktionen in Blöcken
3. Knoten suchen Verifizierungsnummer
4. Gefundene Verifizierungsnummer an alle Knoten senden
5. Block akzeptiert wenn alle Transaktionen valide

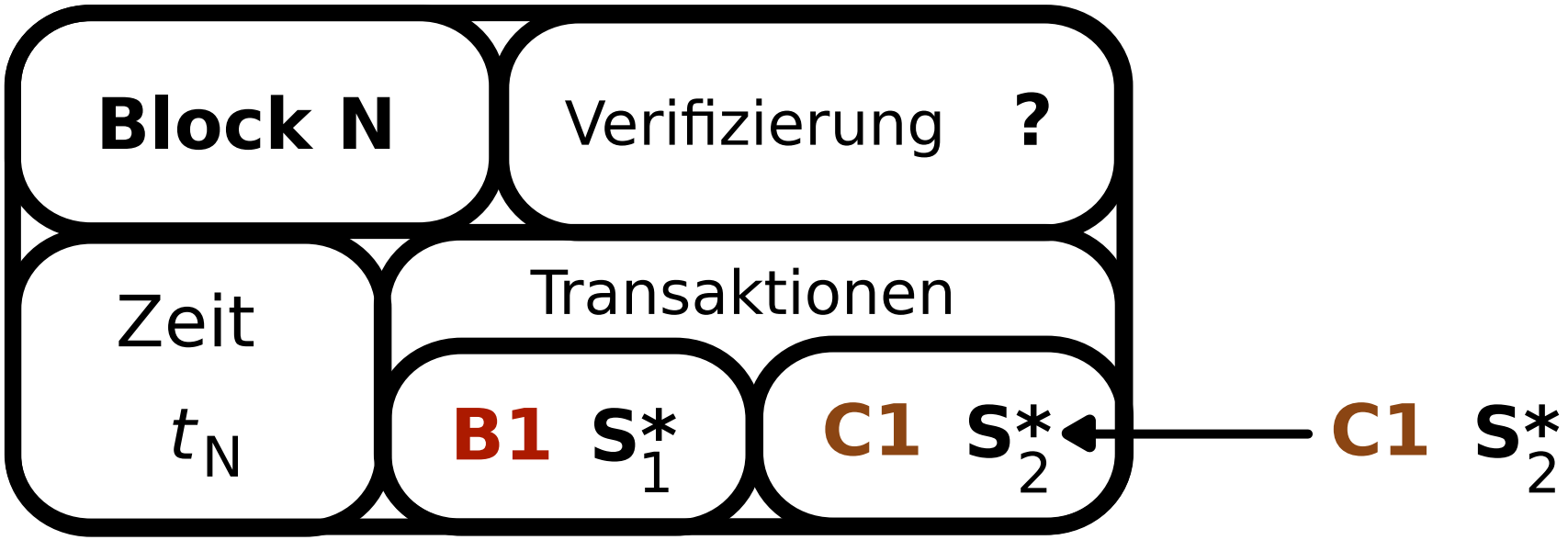
Zusammenfassung: Protokoll

1. **Neue Transaktionen an alle Knoten senden**
2. Sammeln von Transaktionen in Blöcken
3. Knoten suchen Verifizierungsnummer
4. Gefundene Verifizierungsnummer an alle Knoten senden
5. Block akzeptiert wenn alle Transaktionen valide



Zusammenfassung: Protokoll

1. Neue Transaktionen an alle Knoten senden
2. **Sammeln von Transaktionen in Blöcken**
3. Knoten suchen Verifizierungsnummer
4. Gefundene Verifizierungsnummer an alle Knoten senden
5. Block akzeptiert wenn alle Transaktionen valide

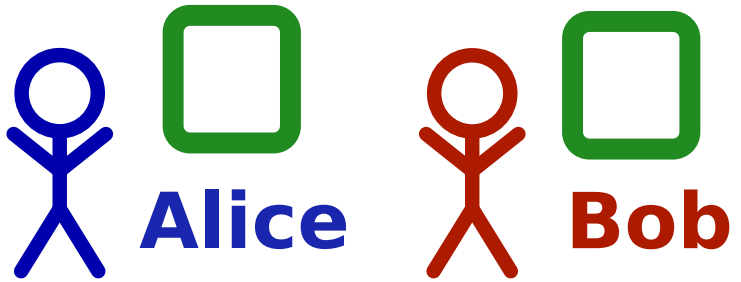


Zusammenfassung: Protokoll

1. Neue Transaktionen an alle Knoten senden
2. Sammeln von Transaktionen in Blöcken
3. **Knoten suchen Verifizierungsnummer**
4. Gefundene Verifizierungsnummer an alle Knoten senden
5. Block akzeptiert wenn alle Transaktionen valide

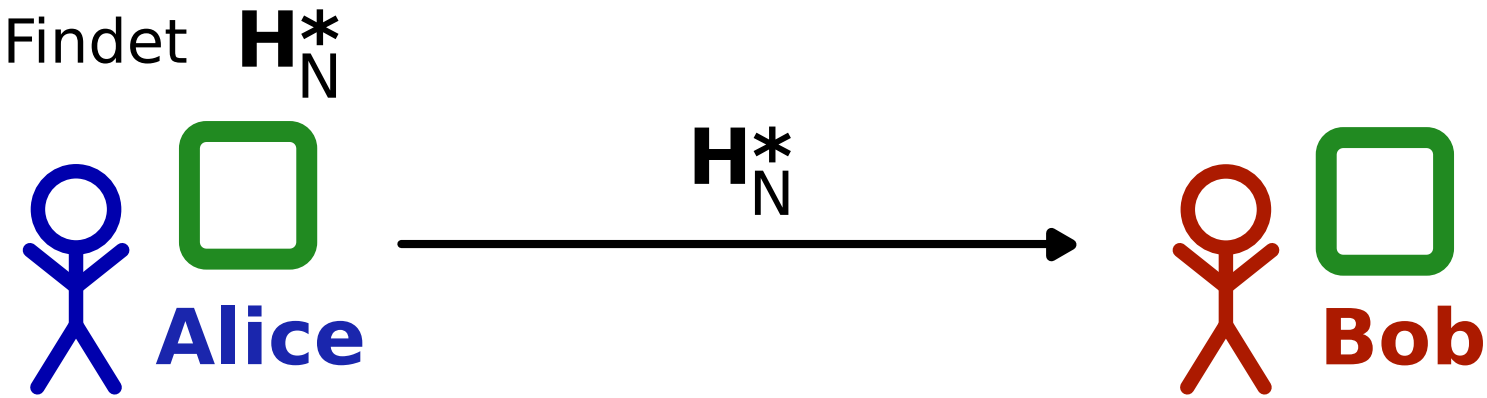
Suchen Verifizierung $\mathbf{H}^*_N$

Verifizierung **?**



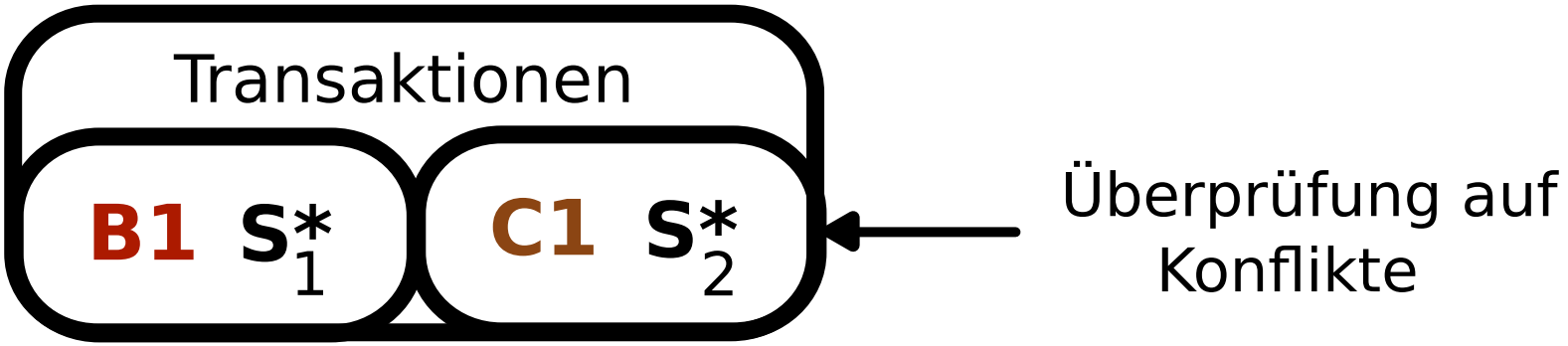
Zusammenfassung: Protokoll

1. Neue Transaktionen an alle Knoten senden
2. Sammeln von Transaktionen in Blöcken
3. Knoten suchen Verifizierungsnummer
4. **Gefundene Verifizierung an alle Knoten senden**
5. Block akzeptiert wenn alle Transaktionen valide



Zusammenfassung: Protokoll

1. Neue Transaktionen an alle Knoten senden
2. Sammeln von Transaktionen in Blöcken
3. Knoten suchen Verifizierungsnummer
4. Gefundene Verifizierung an alle Knoten senden
5. **Block akzeptiert wenn alle Transaktionen valide**



Demonstration

Live Status

- https://blockchain.info/de
- https://bitnodes.earn.com/
- https://bitinfocharts.com/de/markets/#EUR

Geschichte: Bitcoin

Meilensteine

- "Bitcoin: A Peer-to-Peer Electronic Cash System"
  Satoshi Nakamoto, 2008
- 12. Januar 2009: Erste Transaktion
- 2010: Zwei Pizzen für 10.000 BC
- 17. Dezember 2017: 19.783,21$

Vielen Dank für Ihre Aufmerksamkeit!