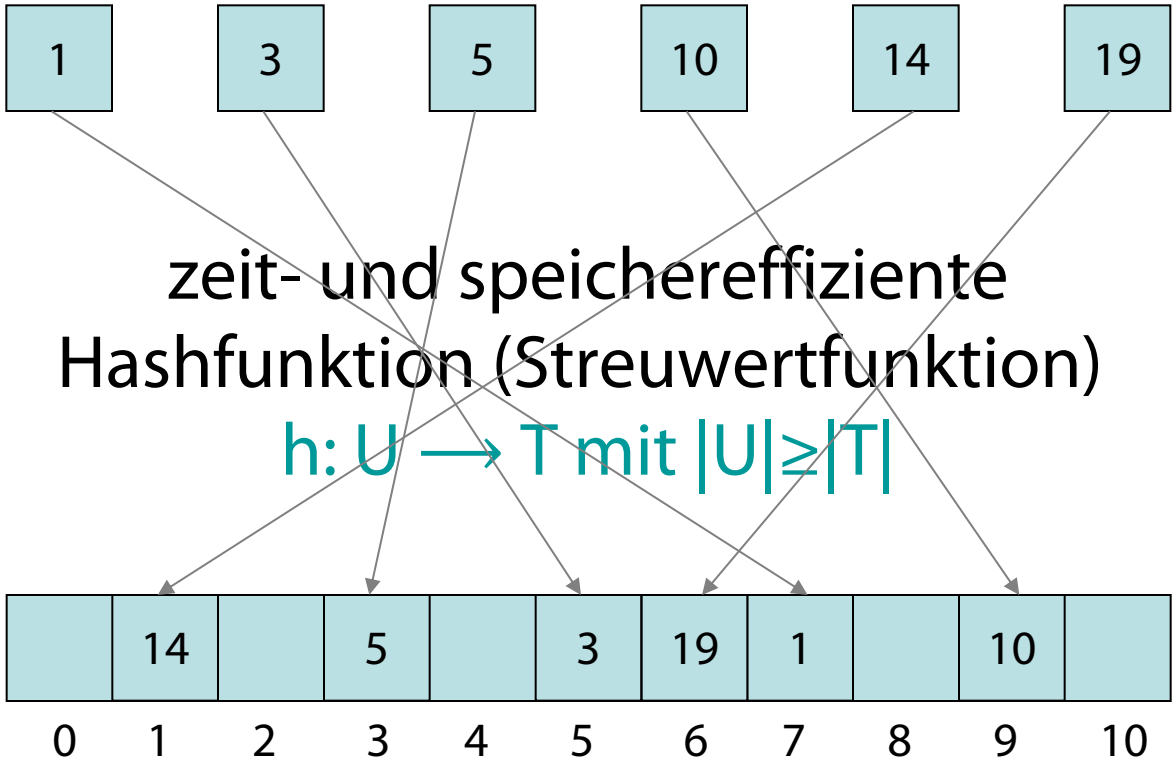
# Einführung in Web und Data Science

## Blockchain Datenmanagement

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

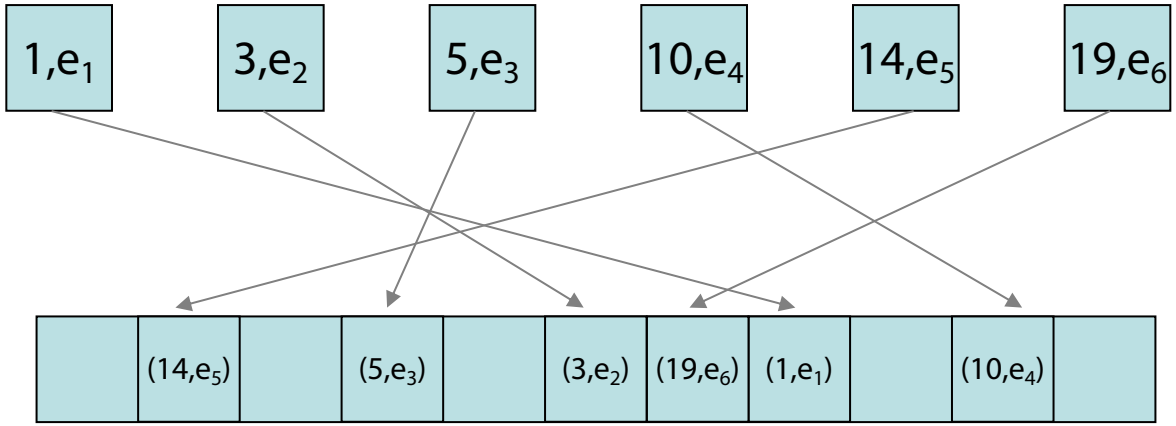
# Hashing (Streuung)

Einige Elemente
aus einer Menge U:



# Vorarbeit: Assoziation durch Hashing

- Schlüssel k seien hier Zahlen aus einem großen Bereich
- Assoziation von k mit e




- Schlüssel k selbst können auch komplexe Objekte sein, es muss nur eine Abbildung auf T definiert sein bzw. werden (Dictionary-spezifische Hash-Funktionen)

# Hashing: Übliches Anwendungsszenario

- Menge U der potentiellen Schlüssel ist „groß“
- Anzahl der Feldelemente length(T) „klein“
- D.h.: |U| >> length(T), aber nur „wenige“ u ∈ U werden tatsächlich betrachtet
- Verschiedene Schlüssel möglicherweise auf gleichen Index abgebildet (Kollision)
- Wenn length(T) genügend groß, sind Kollisionen selten

# Vorarbeit: Hashfunktionen

- Hashfunktionen müssen i.A. anwendungsspezifisch definiert werden (oft für Basisdatentypen Standardimplementierungen angeboten)
- Hashwerte sollen möglichst gleichmäßig gestreut werden (sonst Kollisionen vorprogrammiert)
- Ein erstes Beispiel für U = Integer:

```
function h(u)
    return u mod m
```

wobei m=length(T)

**Falls m keine Primzahl:**
Schlüssel seien alle Vielfache von 10 und Tabellengröße m sei 100
→ Viele Kollisionen

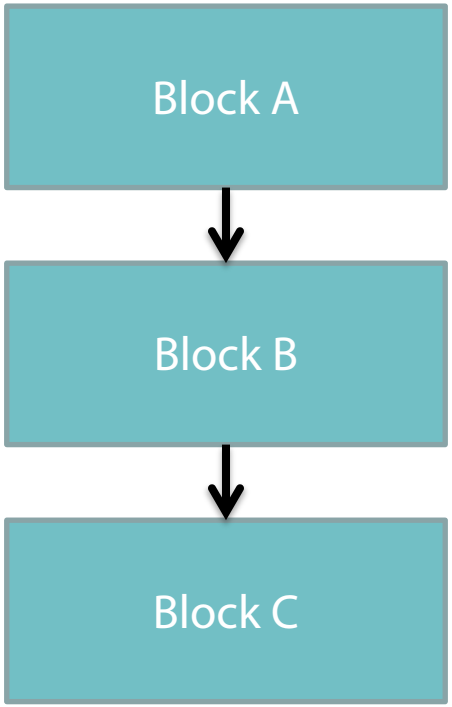
- Die Funktion h kann für beliebige Datentypen definiert werden
- Kleine Veränderung von u heißt große Änderung von u mod m

# Was ist eine Blockchain?

Wie der Name schon sagt, eine **Kette** von **Blöcken**

Was enthalten die **Blöcke**?

- Prinzipiell alle möglichen Daten
  - Währungen
  - Verträge
  - Grundbücher
  - Wikipedia Einträge
  - Stammbäume




Was ist mit der **Kette**?

- Dokumentiert eine Erweiterung der Daten

**Ziel**: **Konsistente** und **nachvollziehbare** Speicherung der Erweiterung von Daten!

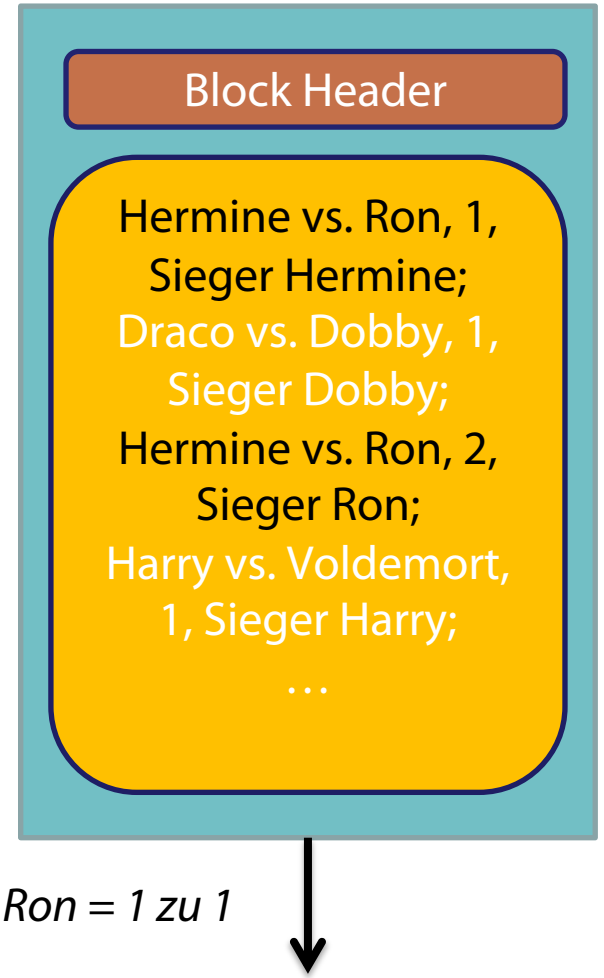
# Beispiel Zaubererduell

Was enthält unserer **Block**?

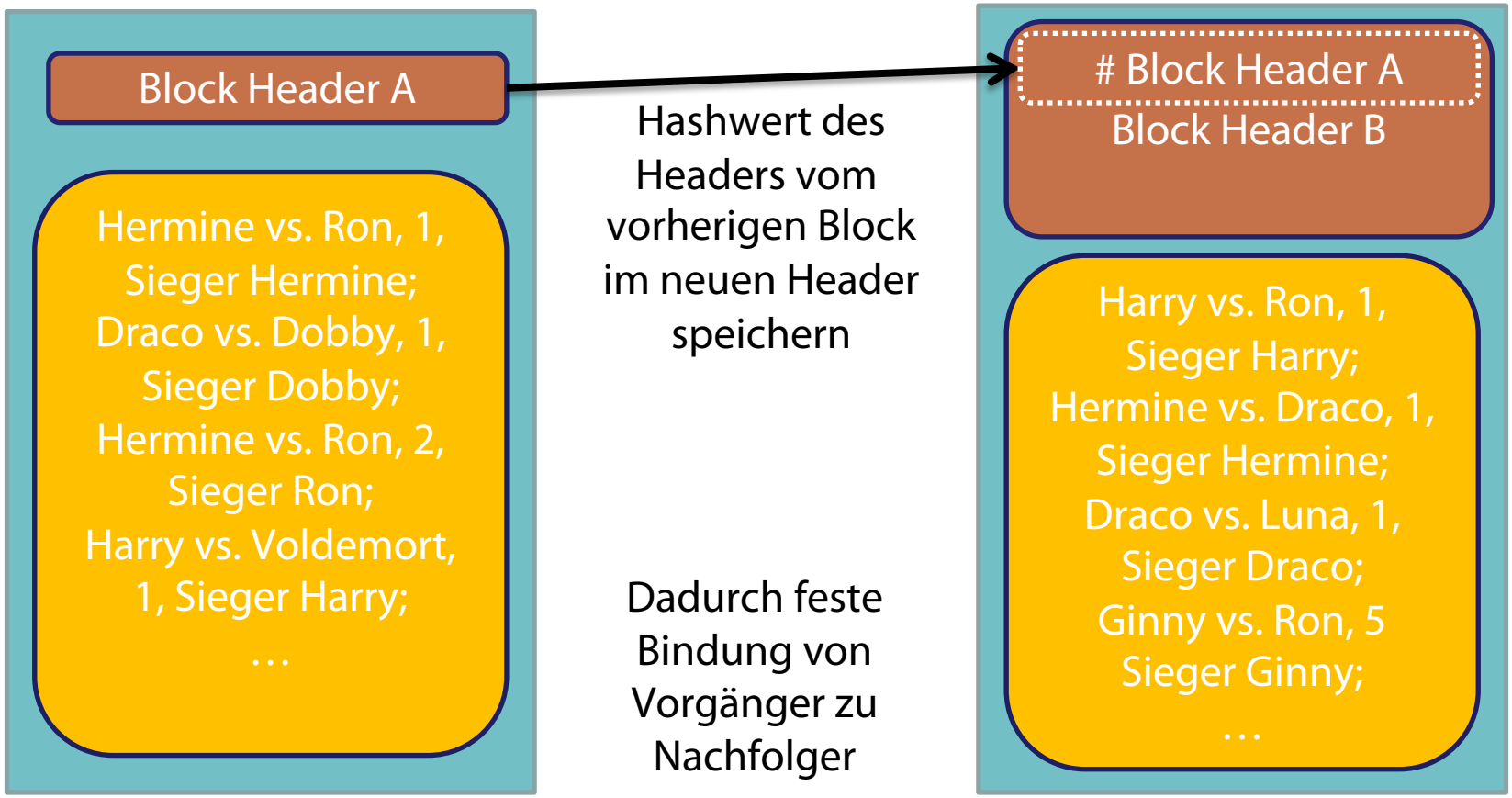
- Duelle zwischen 2 Zaubernden
- Sieger des Duells
- (Duellant A, Duellant B, DuellNr. zwischen Beteiligten, Sieger)

Was können wir damit **nachweisen**?

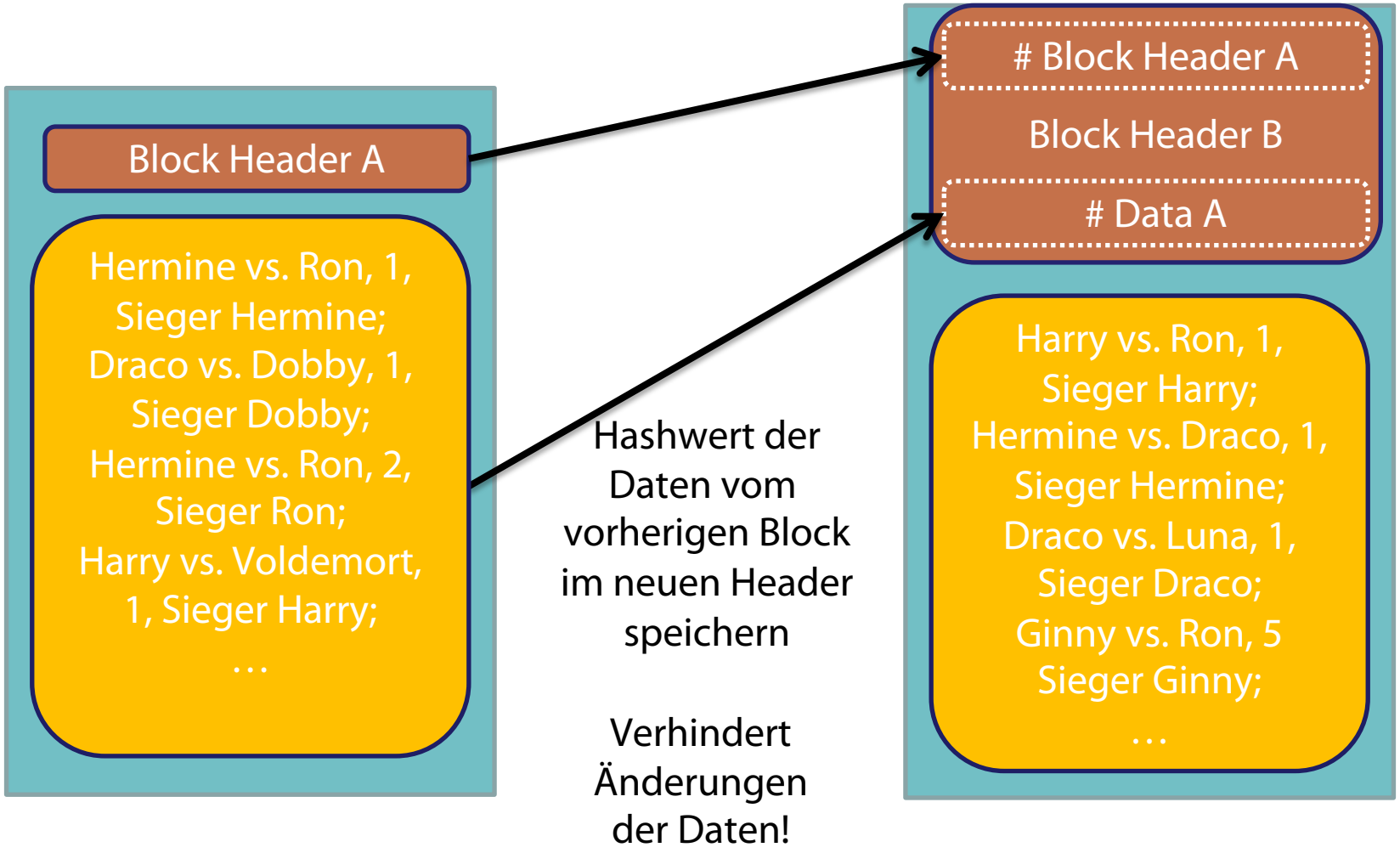
- Wer gegen Wen
- Häufigkeit Sieger/Verlierer über gesamte Kette



# Wie sind die Blöcke verbunden?



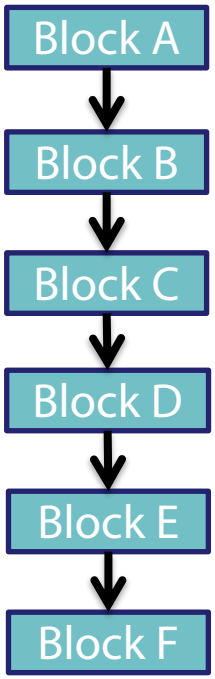
# Wie sind die Blöcke verbunden?



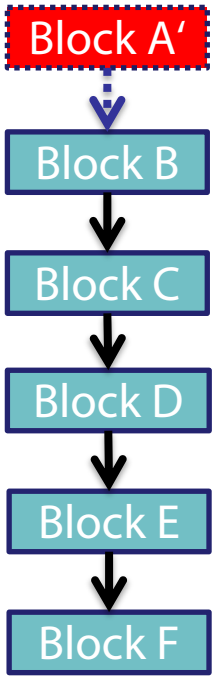
# Kann Draco schummeln?

**Problem:** Verändere die Kette so, dass Draco in der Vergangenheit gewonnen hat.

Aktuelle Kette:




Dracos Kette:




Hashwert anders wegen veränderter Daten!
Block B erkennt seinen Vorgänger nicht an!

Schummeln am **Anfang** oder in der **Mitte** der Kette nicht möglich.

Was ist mit dem **Ende**?

# Wer entscheidet über einen neuen Block?

Prinzipell: **JEDER** Teilnehmer an der Blockchain

Heißt in unserem Fall jeder Zaubernde, **ABER** es muss **Konsens** herrschen!

Wir brauchen somit ein dezentrales **Konsenzverfahren**!
Häufig wird die sog. **Proof-of-Work-**Methode verwendet:

- Aufwendiger Arbeitsnachweis des Blockerstellers
- Einfache Prüfung für jeden anderen Teilnehmer

**BEISPIEL:** Ein neuer Block darf nur erstellt werden, wenn eine Abbildung des Blockinhalts auf einen Zauberspruch genannt werden kann, so dass z.B. der:

- Zauberspruch mit A anfängt,
- aus 2 Worten besteht und
- genau 2 Mal ‚v' enthält

Beispiel: **Avada Kedavra (man muss erst einmal viele Zauberspruchbücher durchsehen, um so eine Abbildung zu finden)**
Einfach zu prüfen, aber Abbildung aufwendig zu finden.

Anforderungen wechseln für jeden neuen Block, da neuer Inhalt gegeben

# Erstellen eines neuen Blocks

1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter
3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer
5. Andere Zauberer **akzeptieren** den **Block** oder stellen **ungültige*** Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt

** Was als ungültig angesehen wird, hängt auch von den Daten ab. In unserem Beispiel :*

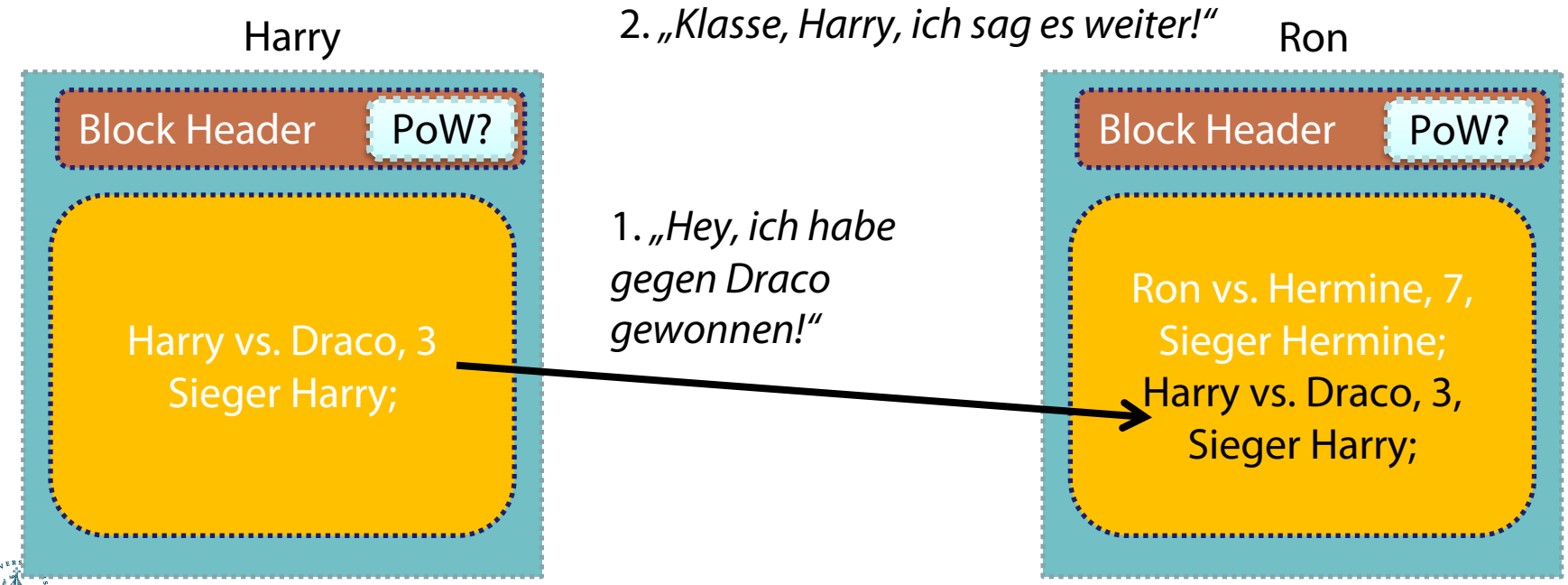
- *Duell gegen sich selbst verboten*
- *Duell mit 2 Siegern/Verlierern (keine doppelte Version eines Duells)*
- *Lücke/Reihenfolge falsch (Duell 3 vor 2, Wo ist Duell 1?)*

Schritte aus https://bitcoin.org/bitcoin.pdf

# Erstellen eines neuen Blocks

1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter

3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer
5. Andere Zauberer **akzeptieren** den **Block** oder stellen ungültige Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt

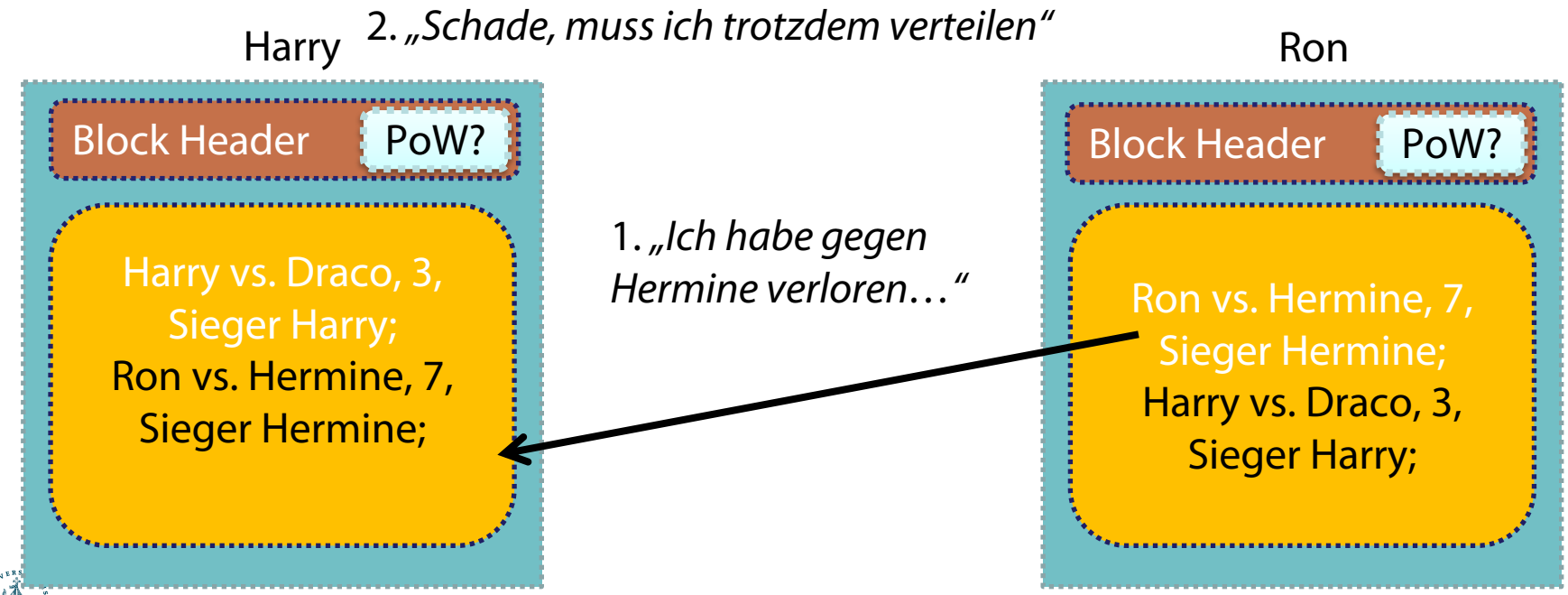



Schritte aus https://bitcoin.org/bitcoin.pdf

# Erstellen eines neuen Blocks

1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter

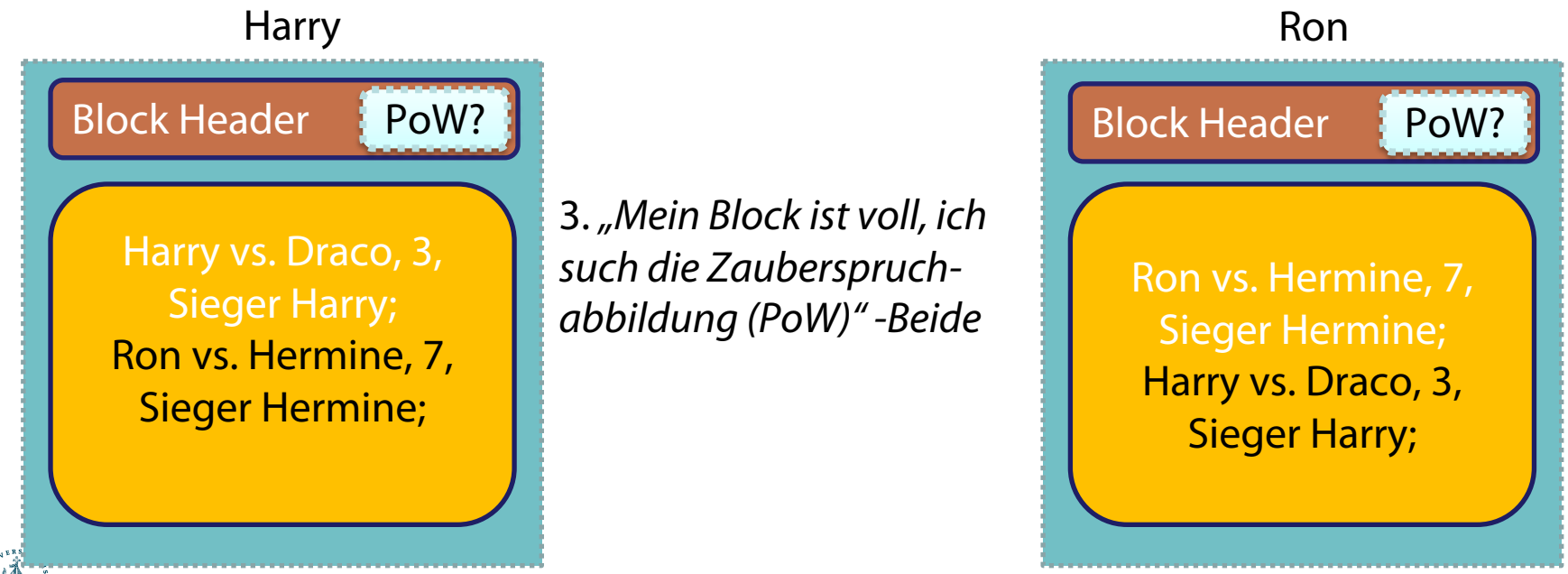
3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer
5. Andere Zauberer **akzeptieren** den **Block** oder stellen invalide Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt




Schritte aus https://bitcoin.org/bitcoin.pdf

# Erstellen eines neuen Blocks

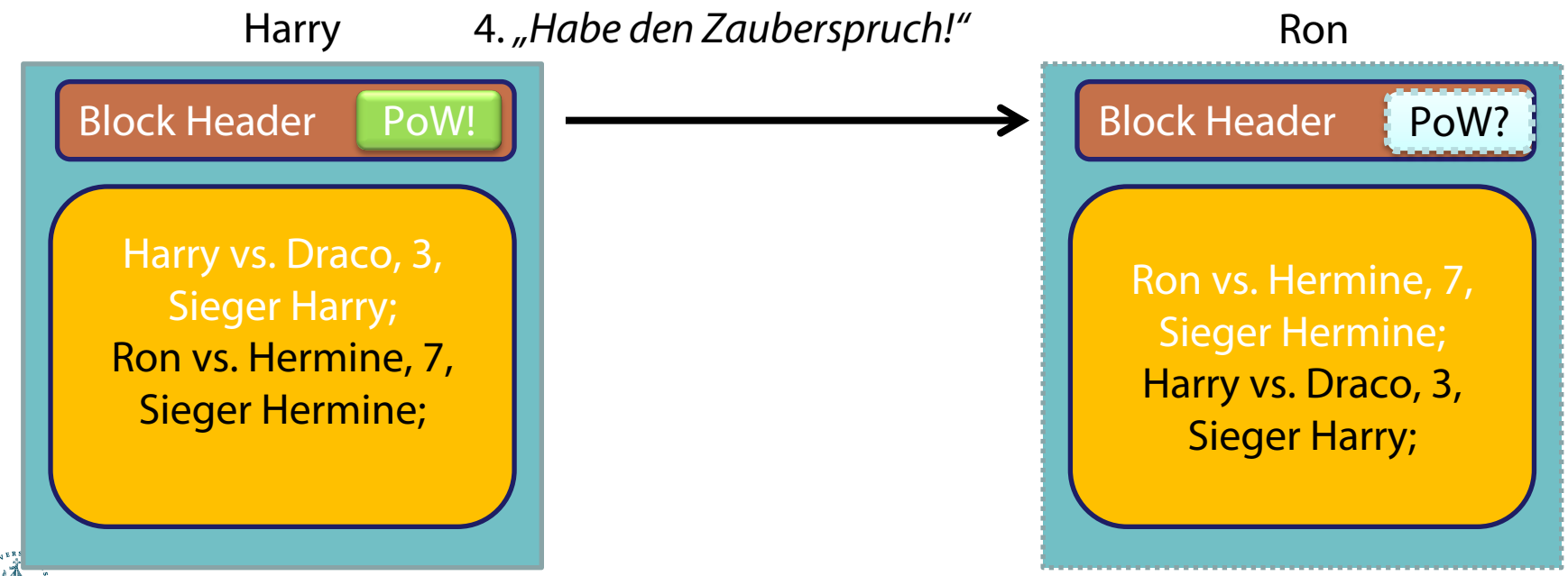
1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter
3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer
5. Andere Zauberer **akzeptieren** den **Block** oder stellen invalide Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt




Schritte aus https://bitcoin.org/bitcoin.pdf

# Erstellen eines neuen Blocks

1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter
3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer
5. Andere Zauberer **akzeptieren** den **Block** oder stellen invalide Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt




Schritte aus https://bitcoin.org/bitcoin.pdf

# Erstellen eines neuen Blocks

1. **Verteile** alle neuen **Daten (bsp. [Harry, Draco, 3, Harry])** an alle bekannten Zauberer
2. Zauberer sammeln **Daten** in Blöcke und verteilen neue **Daten** weiter
3. Zauberer arbeiten am **Proof of Work**, um ihren Block zu veröffentlichen
4. Falls **Proof of Work** erfüllt, **verteile Block** an alle anderen bekannten Zauberer

5. Andere Zauberer **akzeptieren** den **Block** oder stellen invalide Daten fest
6. **Akzeptanz** wird durch Erstellung eines **Nachfolgeblocks** an den neuen Block ausgedrückt


Schritte aus https://bitcoin.org/bitcoin.pdf

# Wie sieht ein realer Proof of Work aus?

Es wird ein Ziel festgelegt, das vom Hashwert erfüllt werden muss,
**beispielweise**:

*Hashfunktion(Header, Daten, RandomNumber) < Ziel*

Da **Header** und **Daten** vorgegeben sind, kann nur die **RandomNumber** verändert werden.

Nach längerem ‚**Raten**' kann eine Lösung gefunden werden.

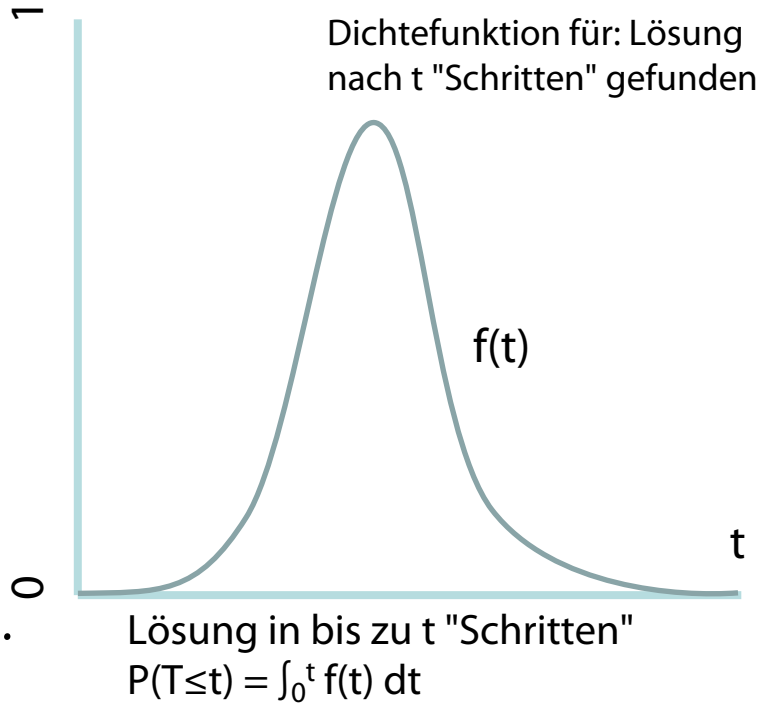
*Hashfunktion(4EF…, 3GH…, **1**) < 100…*

*Hashfunktion(4EF…, 3GH…, **2**) < 100…*

*Hashfunktion(4EF…, 3GH…, **3**) < 100…*

…

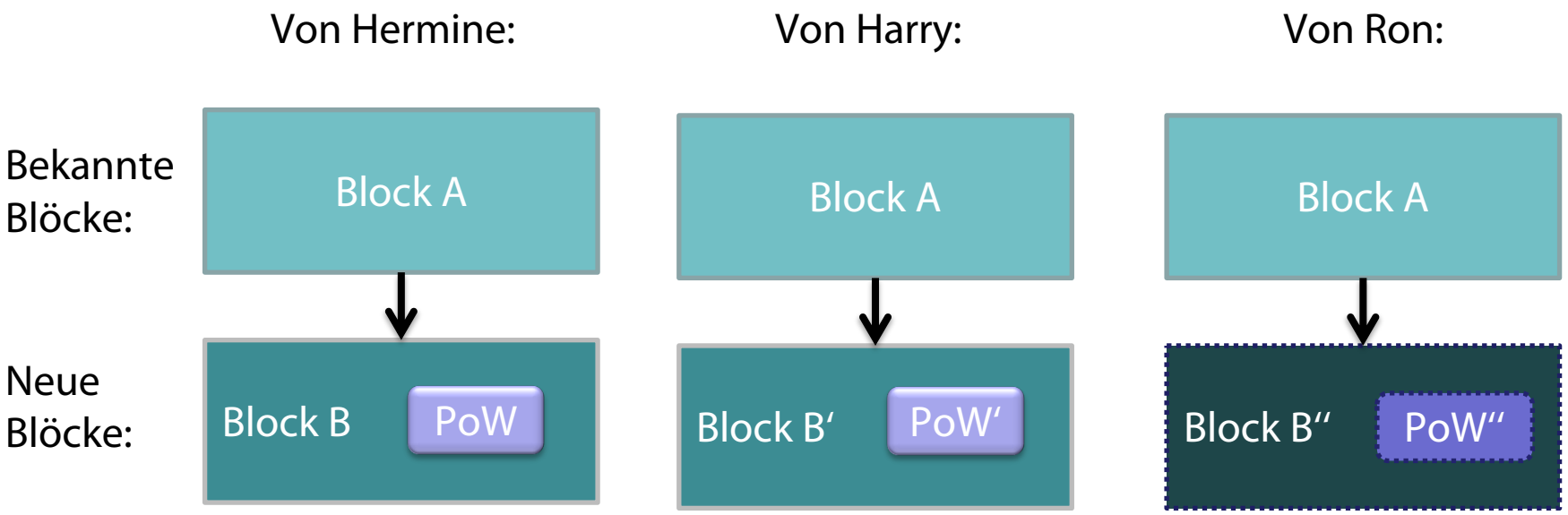
*Hashfunktion(4EF…, 3GH…, **578.321**) < 100…*



# Welche Ketten werden übernommen?

**Problem:** Zauberer sind verteilt, d.h. Ketten können am Ende variieren, wenn viele Zaubernde einen neuen Block erstellen.

Welchen Block soll Ron wählen?

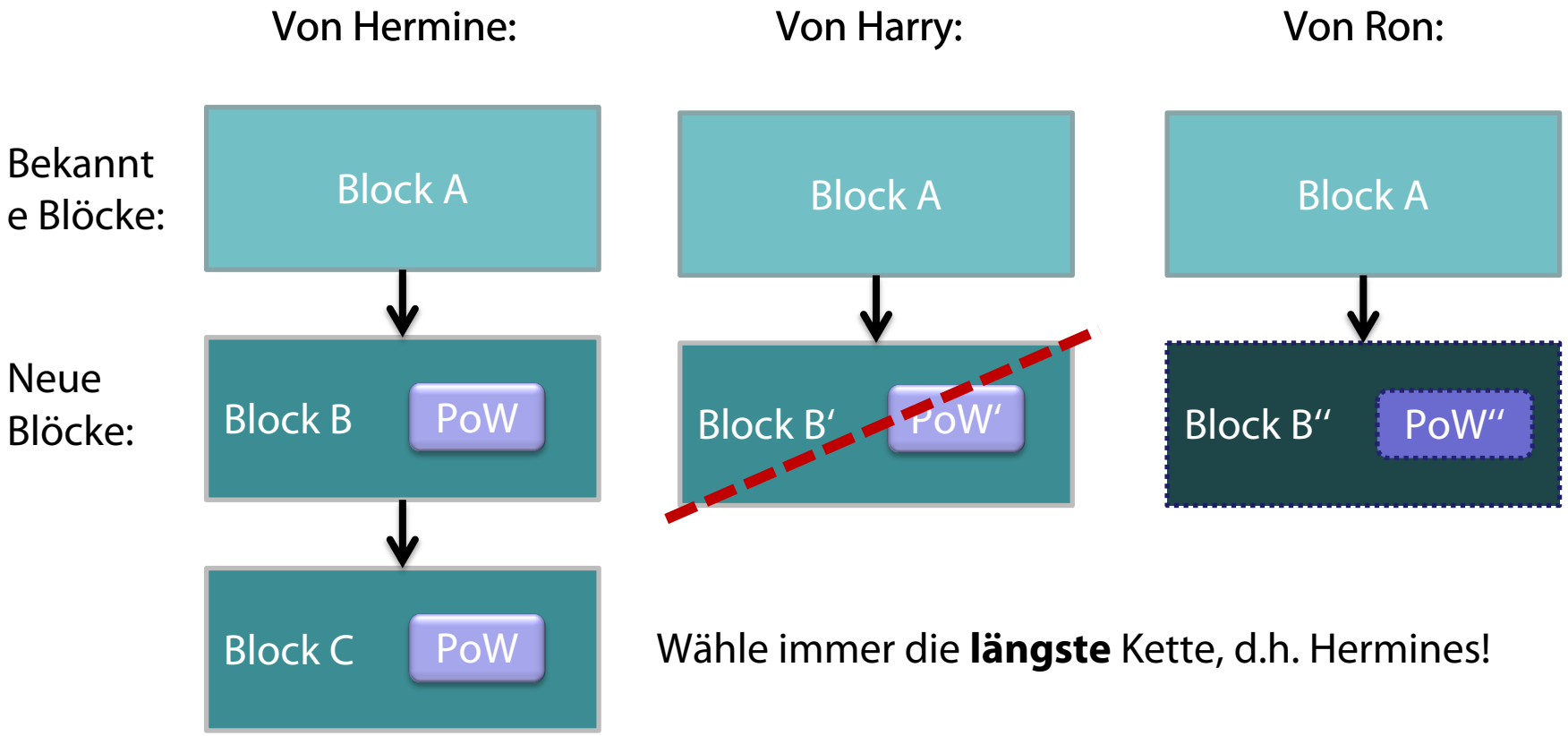



- PoW kann für beide Blöcke getestet werden
- Duelle sind konsistent und Blöcke gefüllt

→ Ron kann sich für einen von beiden entscheiden, denn noch ist keiner wirklich besser als der andere Block

# Welche Ketten werden übernommen?

**Problem:** Zauberer sind verteilt, d.h. die Kette kann am Ende variieren, wenn viele Zaubernde einen neuen Block erstellen.



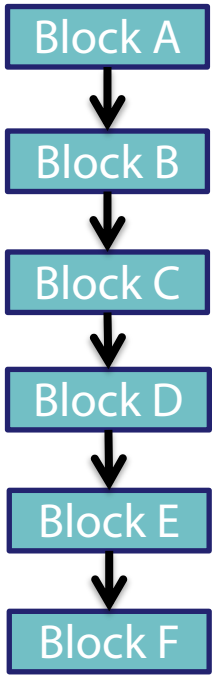

Wähle immer die **längste** Kette, d.h. Hermines!

# Kann Draco schummeln? Revisited

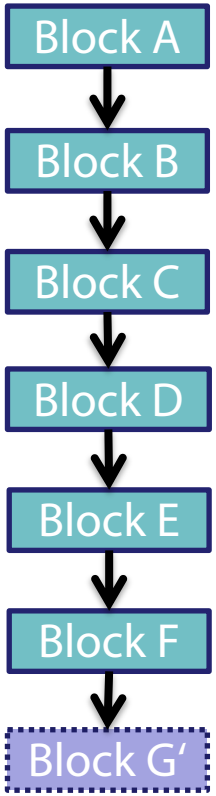
**Draco kämpft gegen Harry, Harry gewinnt. Beide tragen Event in letzten Block ein**

**Problem:** Verändere die Kette so, dass Draco in der Gegenwart gewonnen hat.

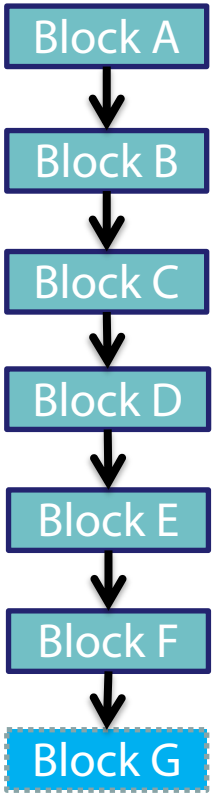
Aktuelle Kette:




Dracos Kette:




Harrys Kette:



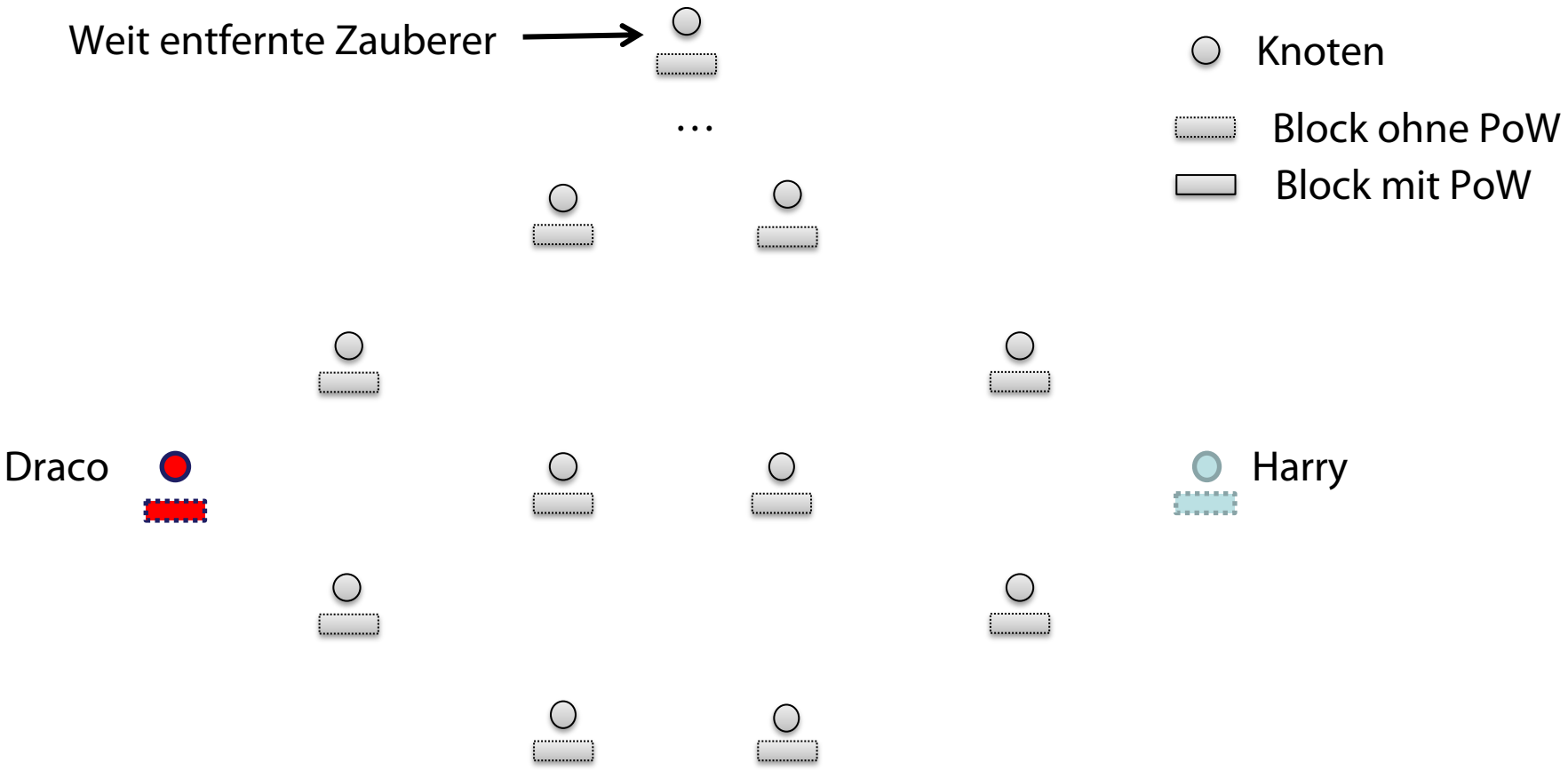

Draco muss erstmal einen PoW leisten, damit er seinen Block verteilen kann.

Draco muss schneller sein als Harry!

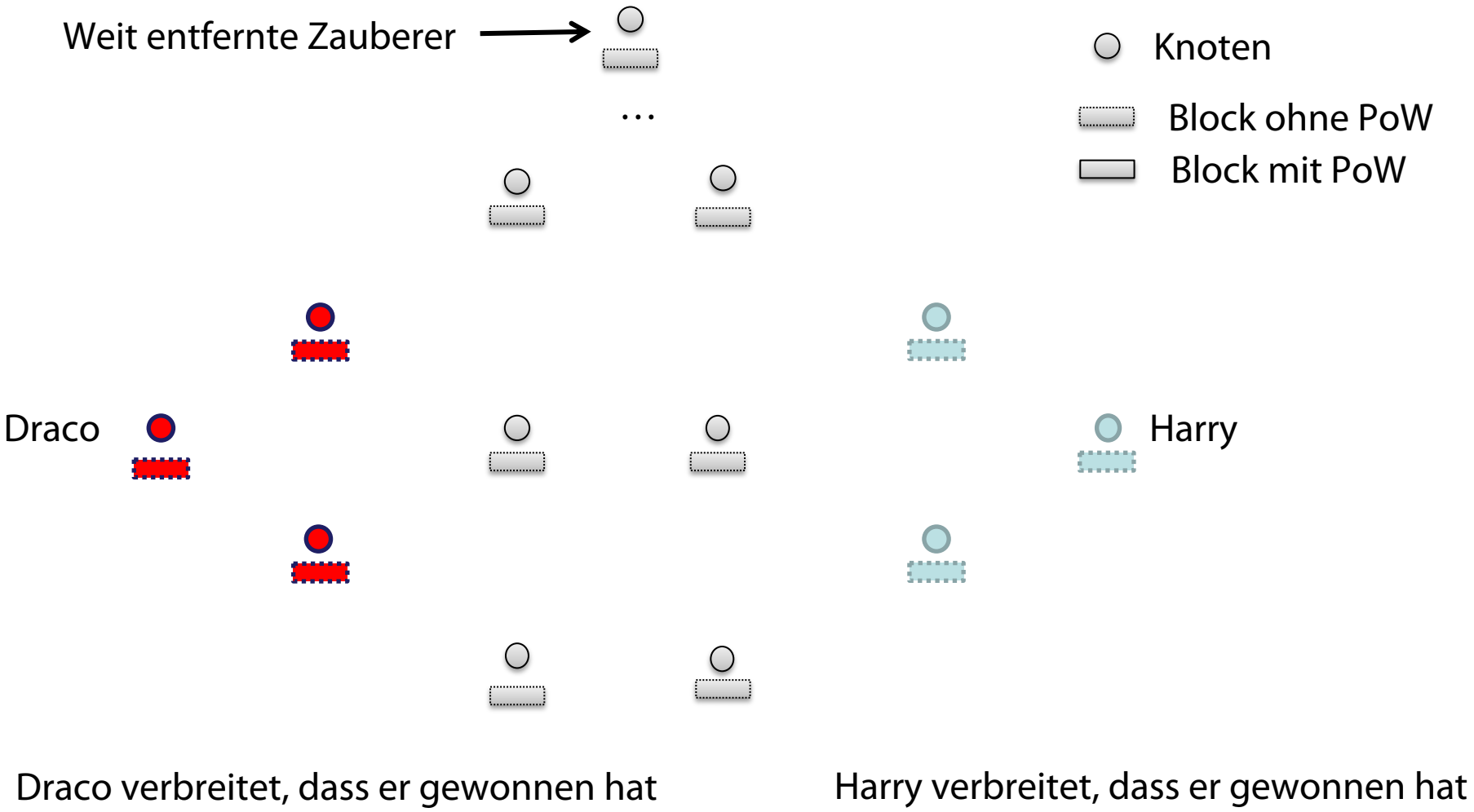
Gewinnchance: 50 zu 50*

*bei gleicher Reichweite im Netzwerk

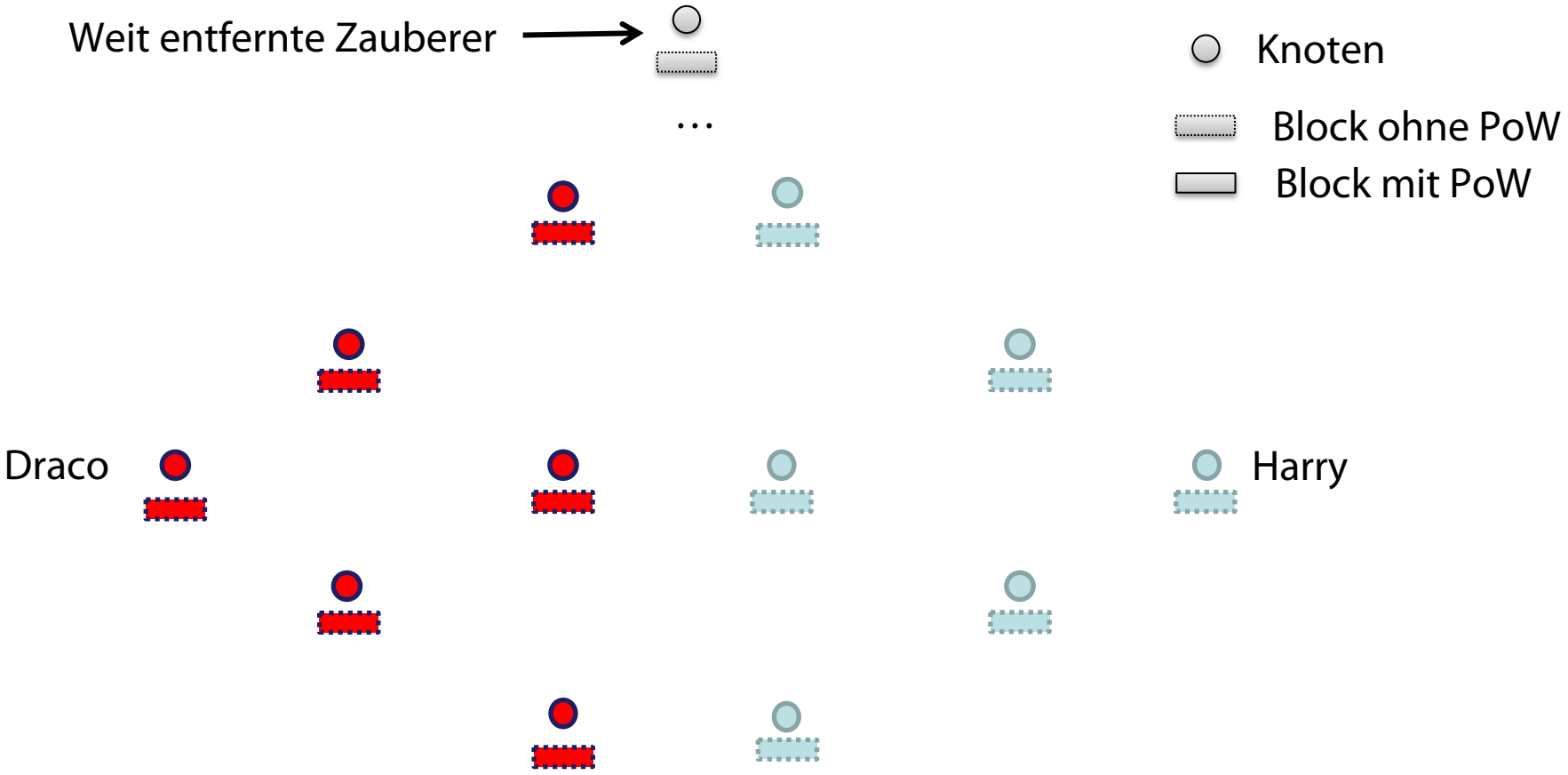
# Kann Draco schummeln? Revisited



# Kann Draco schummeln? Revisited

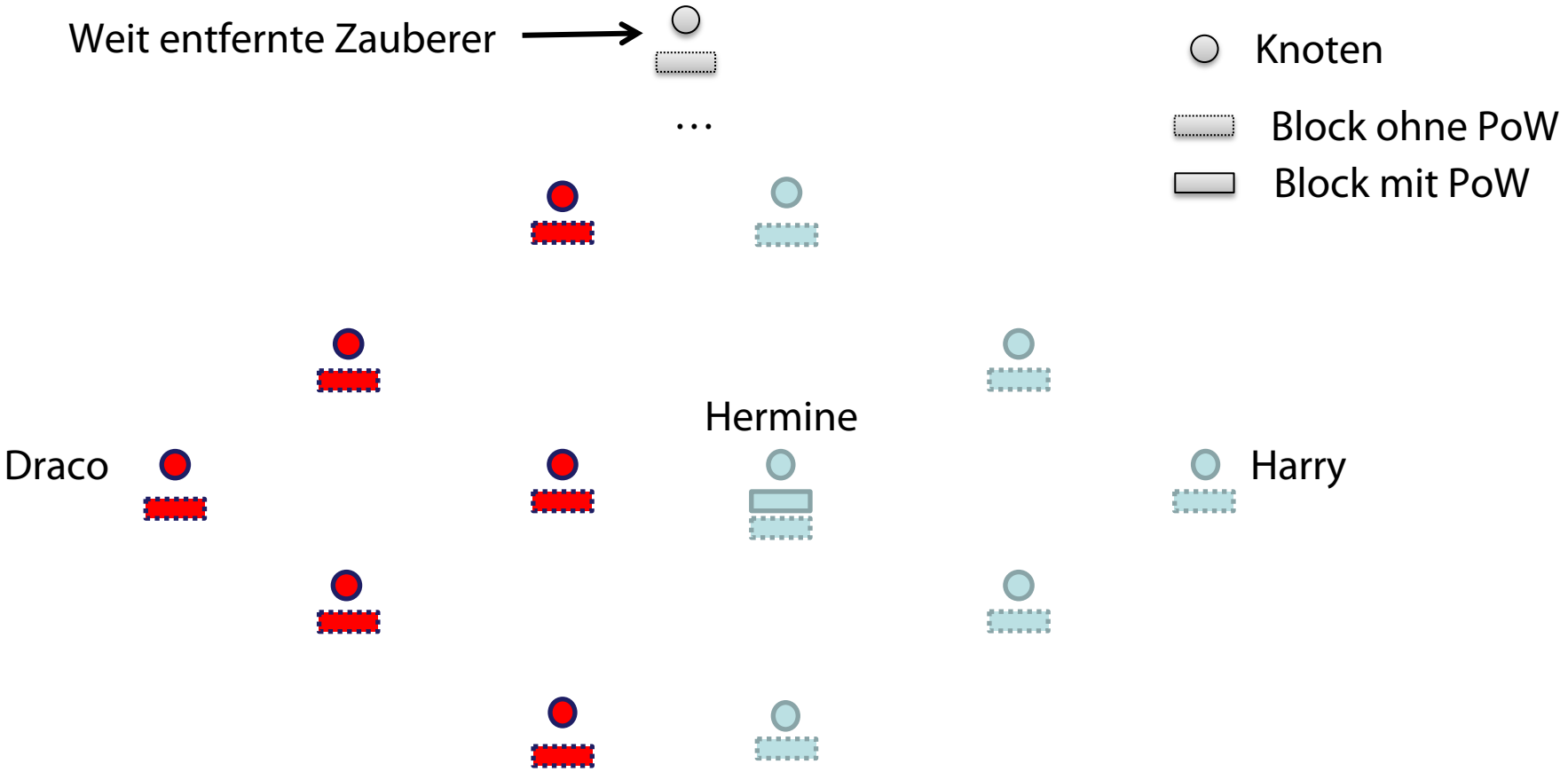


# Kann Draco schummeln? Revisited




Es kann nur einen Sieger pro Duell geben!
Kein Knoten wird beide als Sieger in seinen Block aufnehmen.

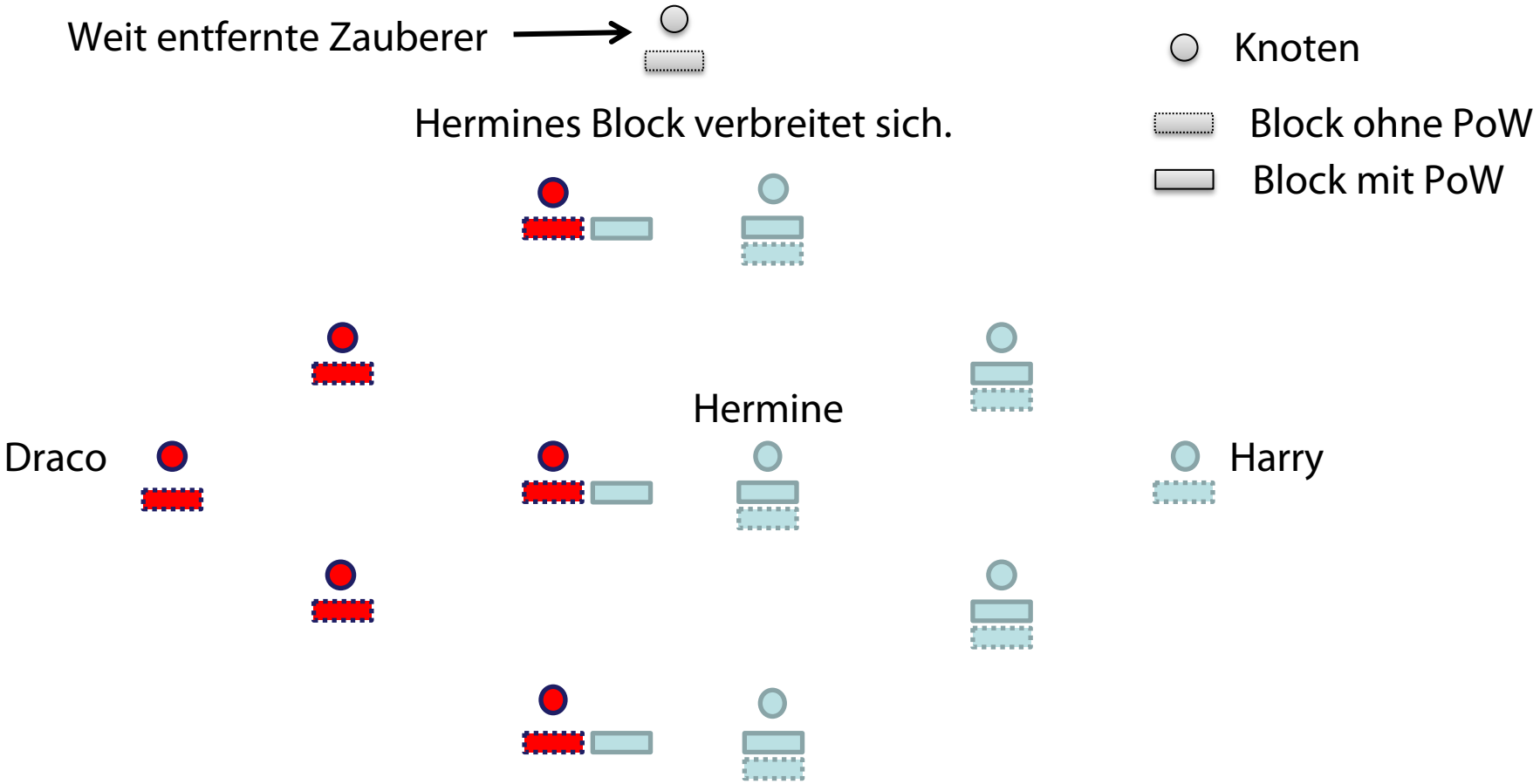
# Kann Draco schummeln? Revisited




Der Knoten Hermine erfüllt den **Proof of Work** für Harrys Version.

Hermine arbeitet nun an Nachfolgeblock

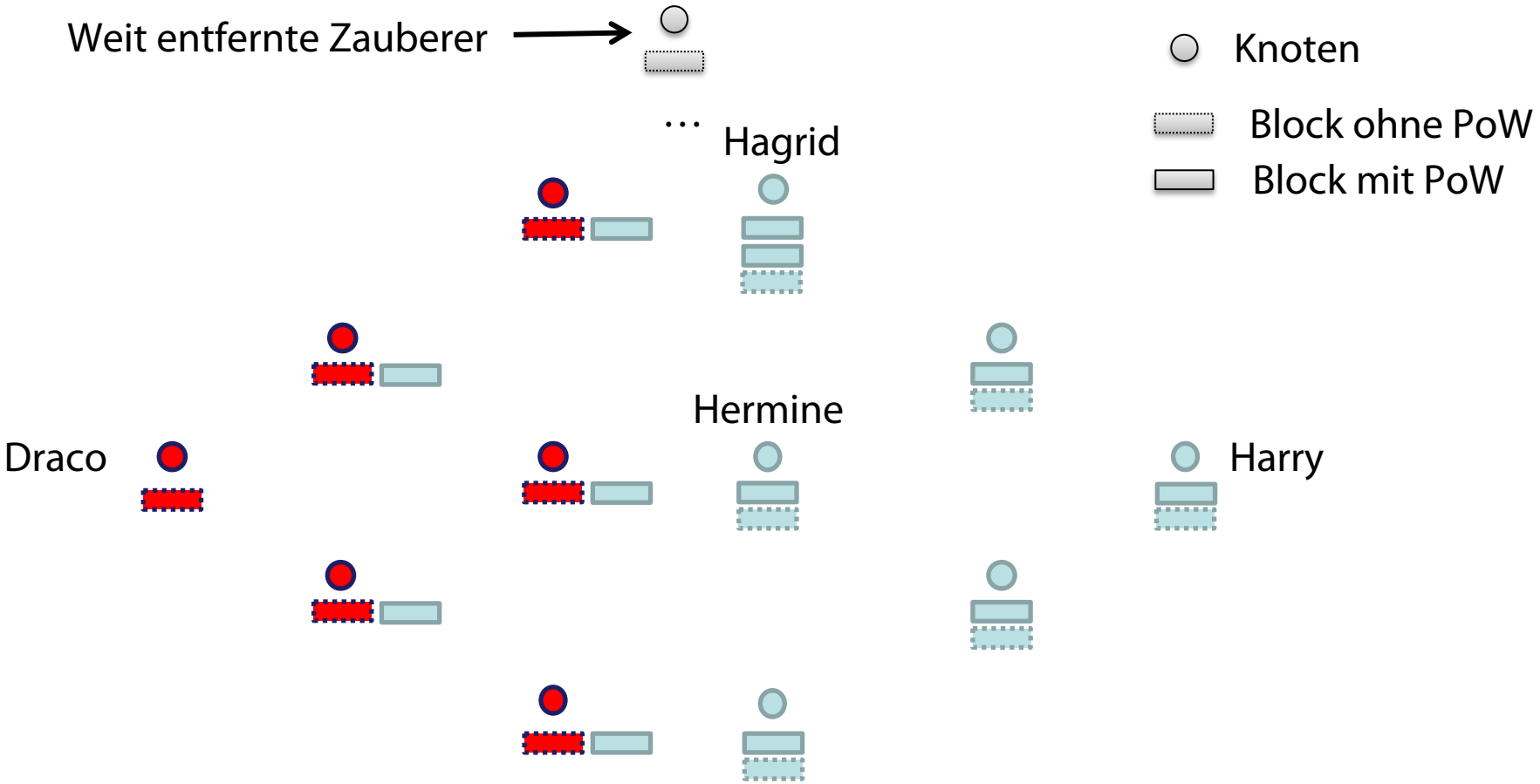
# Kann Draco schummeln? Revisited




Gruppe Draco erhält Hermines Block, akzeptiert diesen aber nicht.

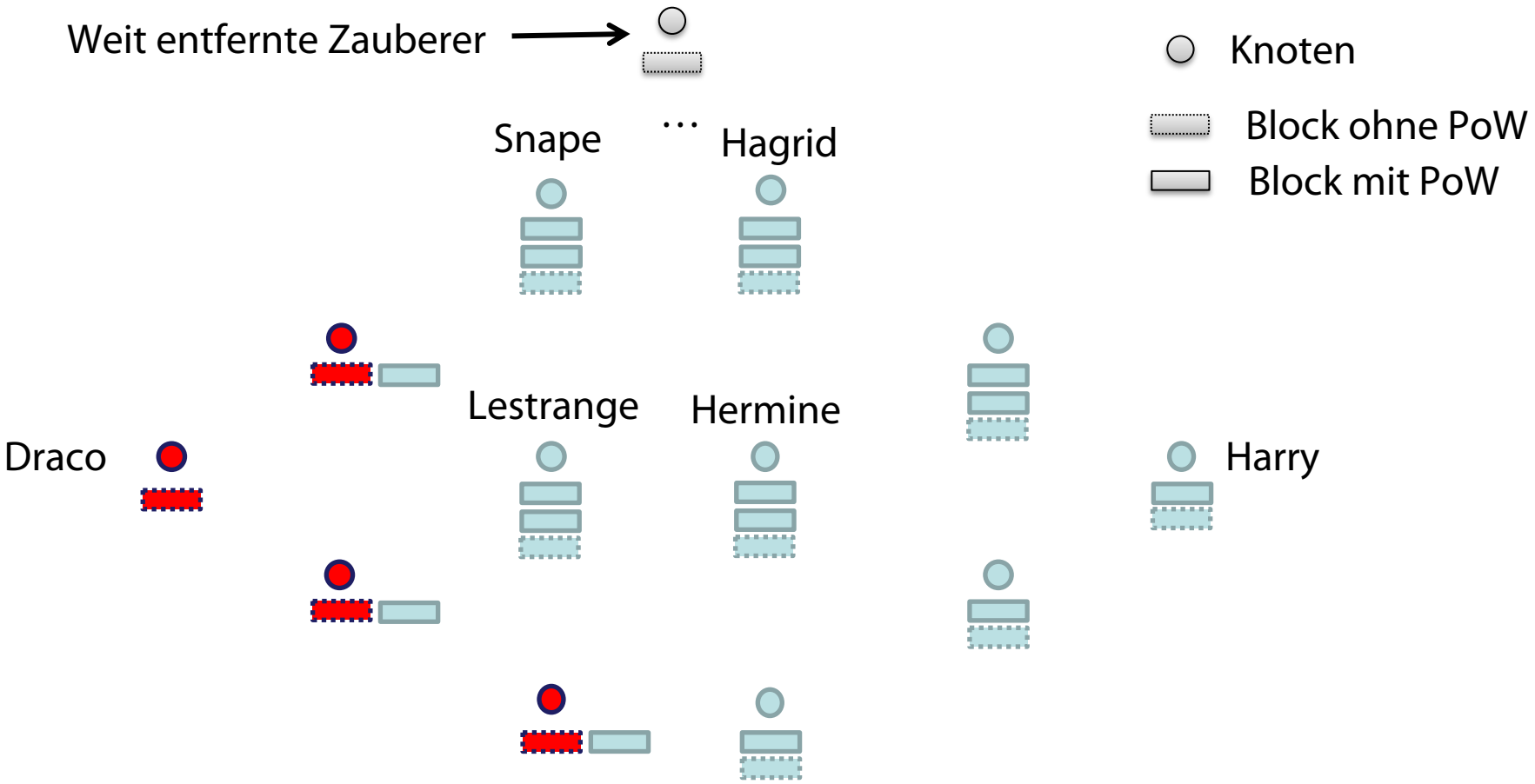
Gruppe Harry arbeitet nun an Nachfolgeblock

# Kann Draco schummeln? Revisited



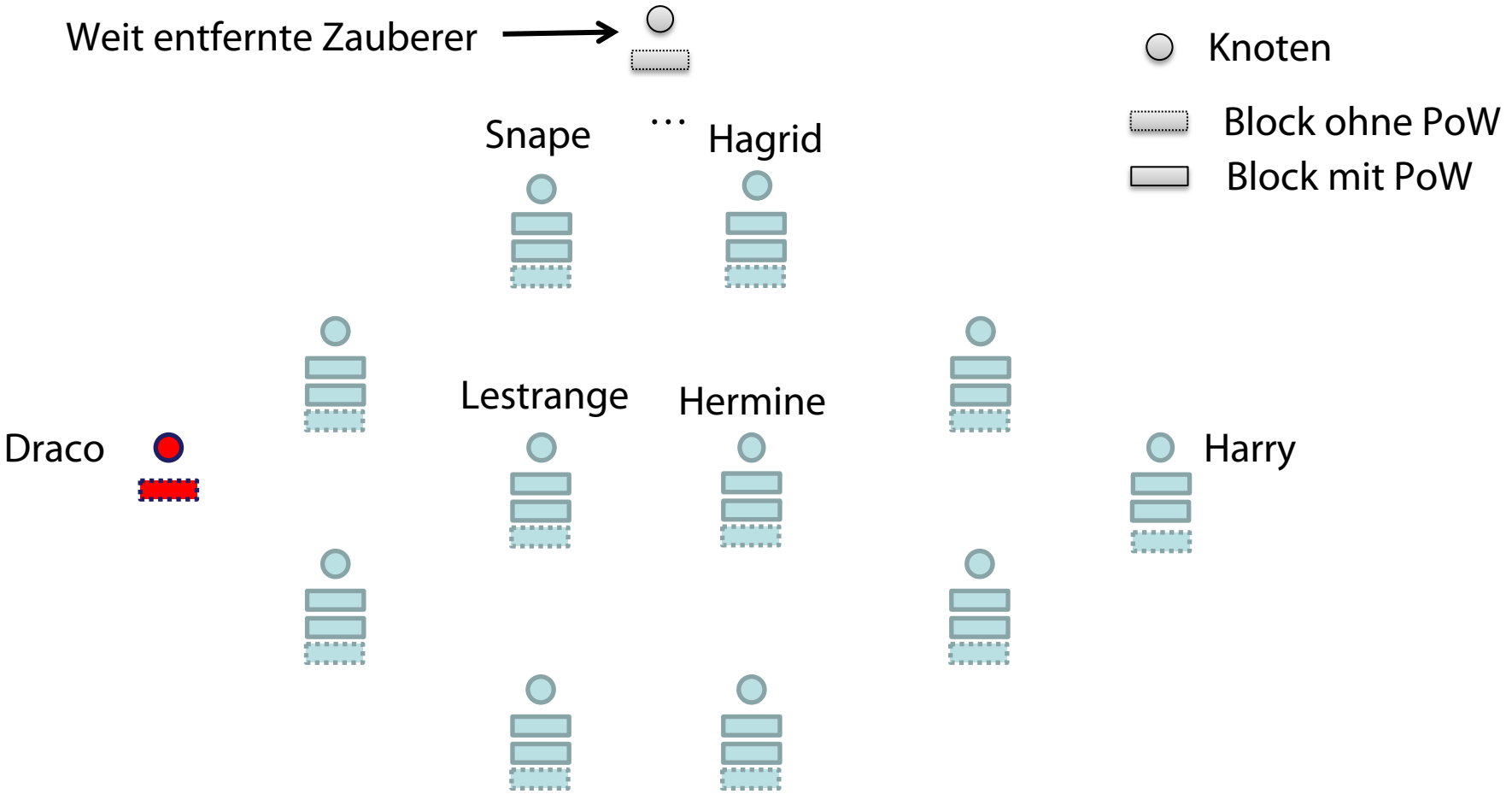

Hagrid erfüllt **Proof of Work** mit einem Block der auf Hermines aufbaut.

# Kann Draco schummeln? Revisited




Snape und Lestrange erkennen längere Kette an und arbeiten nun an der Version in der Harry gewonnen hat.

# Kann Draco schummeln? Revisited

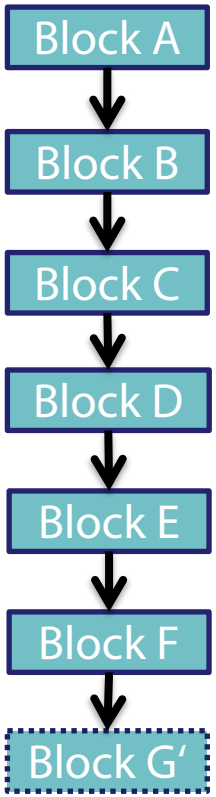



Es sieht schlecht aus für Draco…

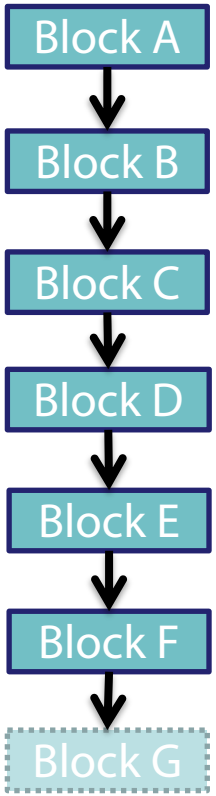
# Kann Draco schummeln? Revisited

**Problem:** Verändere die Kette so, dass Draco in der Gegenwart gewonnen hat.

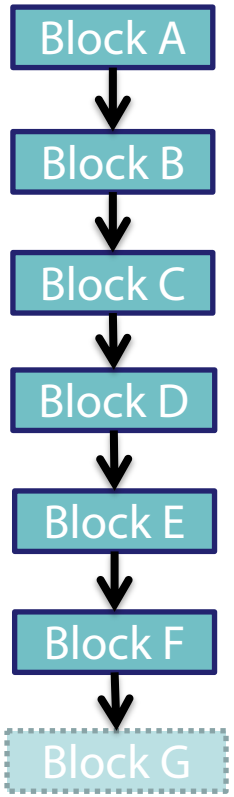
Dracos Kette:




Harrys Kette:




Kette von Dumbledore's Armee:

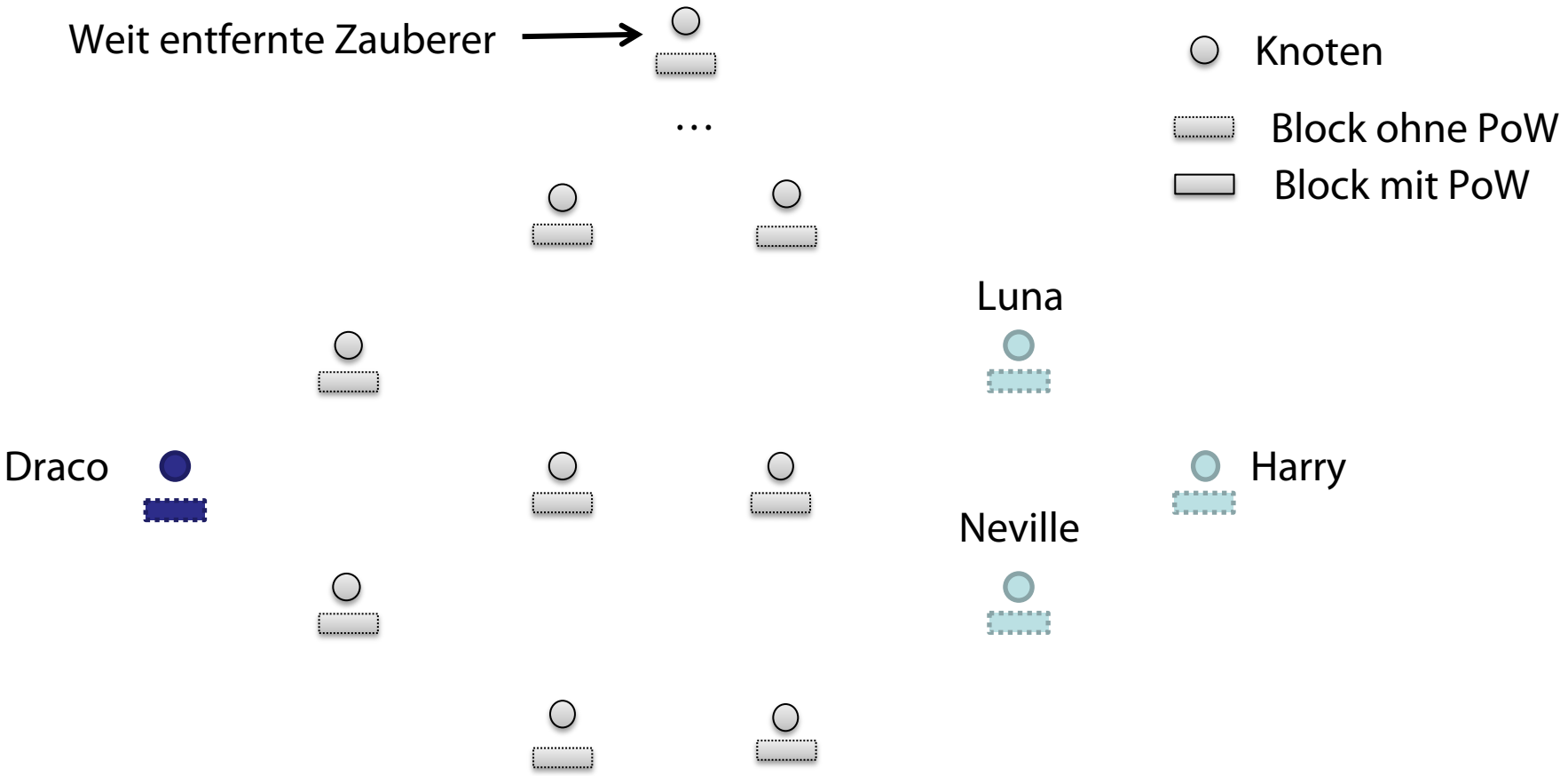



Wahrscheinlichkeit, dass einer der anderen Zauberer oder Harry seine Sicht veröffentlichen kann größer!

Plötzlich Rennen gegen viele andere!

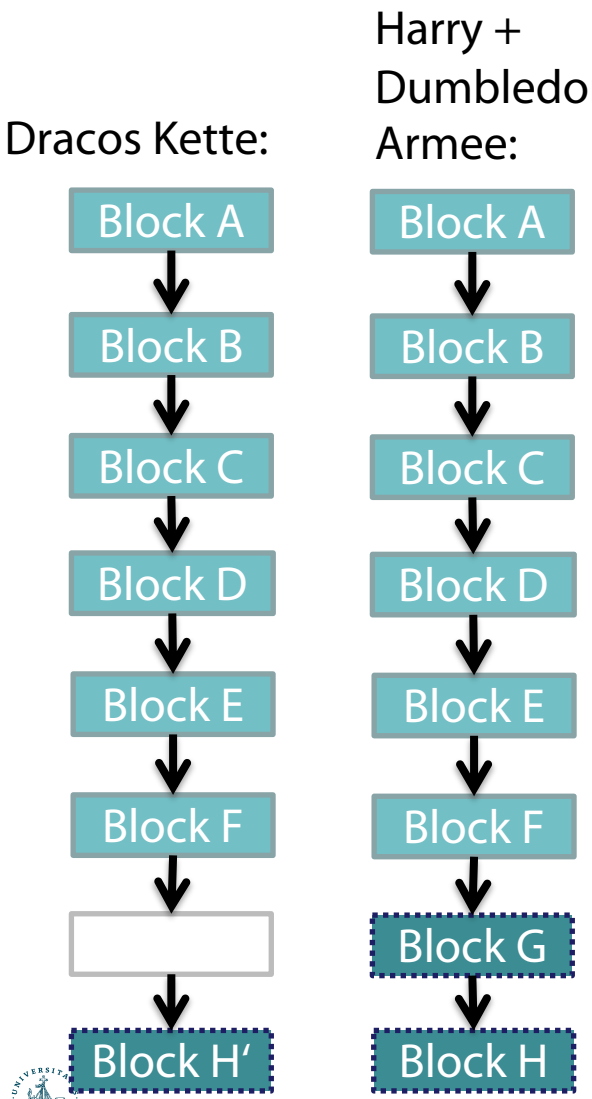
Dracos Chancen, einen neuen Block vor den anderen zu erstellen, sinken mit der Anzahl der anderen Beteiligten.

# Kann Draco schummeln? Revisited




Schon von der Ausgangslage hat Draco ein Problem, hat aber trotzdem noch die Möglichkeit zu gewinnen

# Kann Draco schummeln? Revisited




**Problem:** Verändere die Kette so, dass Draco in der Gegenwart gewonnen hat.

Draco hat trotzdem immer noch eine Chance seinen **Block G‘** zu verteilen.

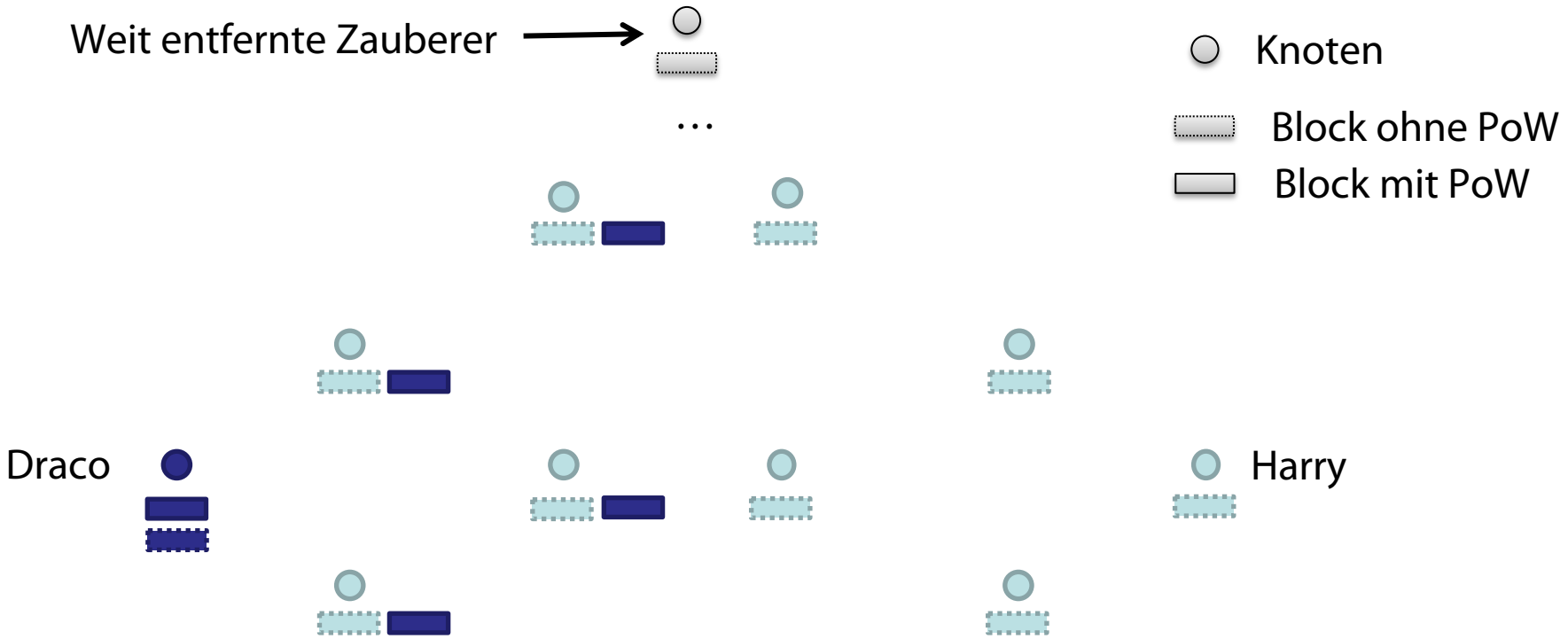
Damit ist der **Block H‘** durch den Hash allerdings anders als der von der anderen Gruppe.
Und Draco muss **erneut** seinen **Proof of Work** für diesen Block alleine leisten.

Die Wahrscheinlichkeit dafür, dass jemand aus Harrys Gruppe den **Proof of Work** für **G** und **H** findet, steigt mit der Anzahl der Teilnehmenden.

Dracos **G‘** würde verworfen, wenn die Blöcke **G** und **H** vor **H‘** veröffentlicht werden.

**Mehrheiten** haben bessere Chancen!

# Kann Draco schummeln? Revisited




Trotz des Vorsprungs muss Draco alleine ein **Proof of Work** für den nächsten Block erfüllen, bevor der Rest umgestimmt wird.

Harrys Gruppe muss nachziehen, hat aber durch die Anzahl eine höhere Chance auf einen **Proof of Work.**

# Zusammenfassung

- Blockchains bestehen aus verketteten Blöcken, die **beliebige** Daten enthalten können.
- **Proof of Work** ermöglicht Konsensverfahren ohne zentrale Instanz und limitiert die neu generierten Blöcke.
- Die **längste** Kette gewinnt.
- Manipulation durch Hashverfahren innerhalb der Liste ausgeschlossen.
- Manipulation am **Ende** möglich, aber nur mit großer Mehrheit (>50%) erfolgsversprechend