Eine Einführung zu CRDTs

Was und warum sind eigentlich
Conflict-free **R**eplicated **D**ata **T**ypes?

Nook 2023

Jakob Runge

Was finde ich hier eigentlich spannend?

- Eine Art und Weise, Daten zu organisieren
- Neu, ✨ und aufregend (~ 2006)
- Komplexität □🎇
- Eine Nische in verteilten Systemen

Anwendungsfälle von CRDTs

- Verteilte Systeme
- Offlinefähige Apps
- Webseiten mit kollaborativen Anteilen

Unabhängige, konfliktierende Veränderungen

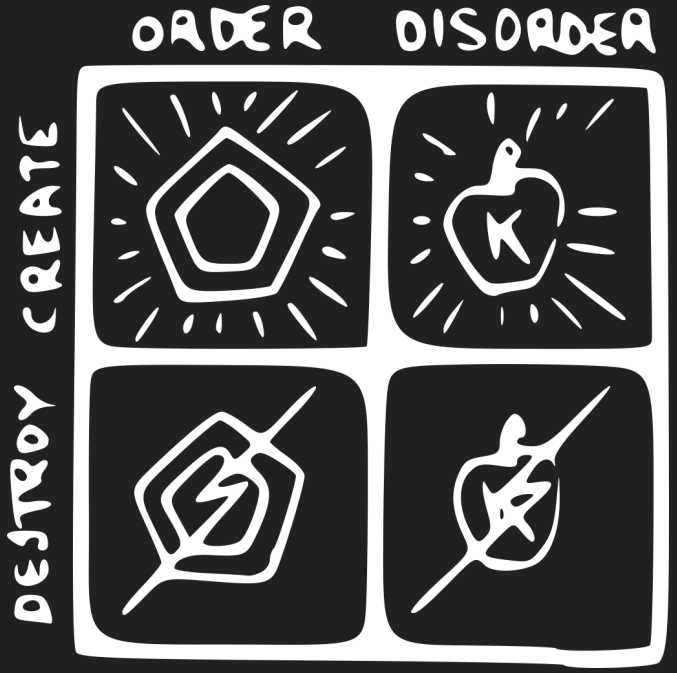
Was wir uns davon versprechen

- Stärkere Unabhängigkeit von Prozessen
- ‘Einfache’ Synchronisation
- Hohe Verfügbarkeit
- Große Flexibilität
  - Anzahl der kooperierenden Prozesse
  - Wahl der Protokolle, APIs, Datenbanken

Abwägen von Komplexität

- Ein Netz hat Löcher
- Eine Software hat Komplexität

~ mehr ≠ besser



https://principiadiscordia.com/book/70.php

Abwägen von Komplexität

- Replication vs. Partitioning
- Datenbanken
- Offline- /Local-First

https://principiadiscordia.com/book/70.php

Replication ≠ Partitioning

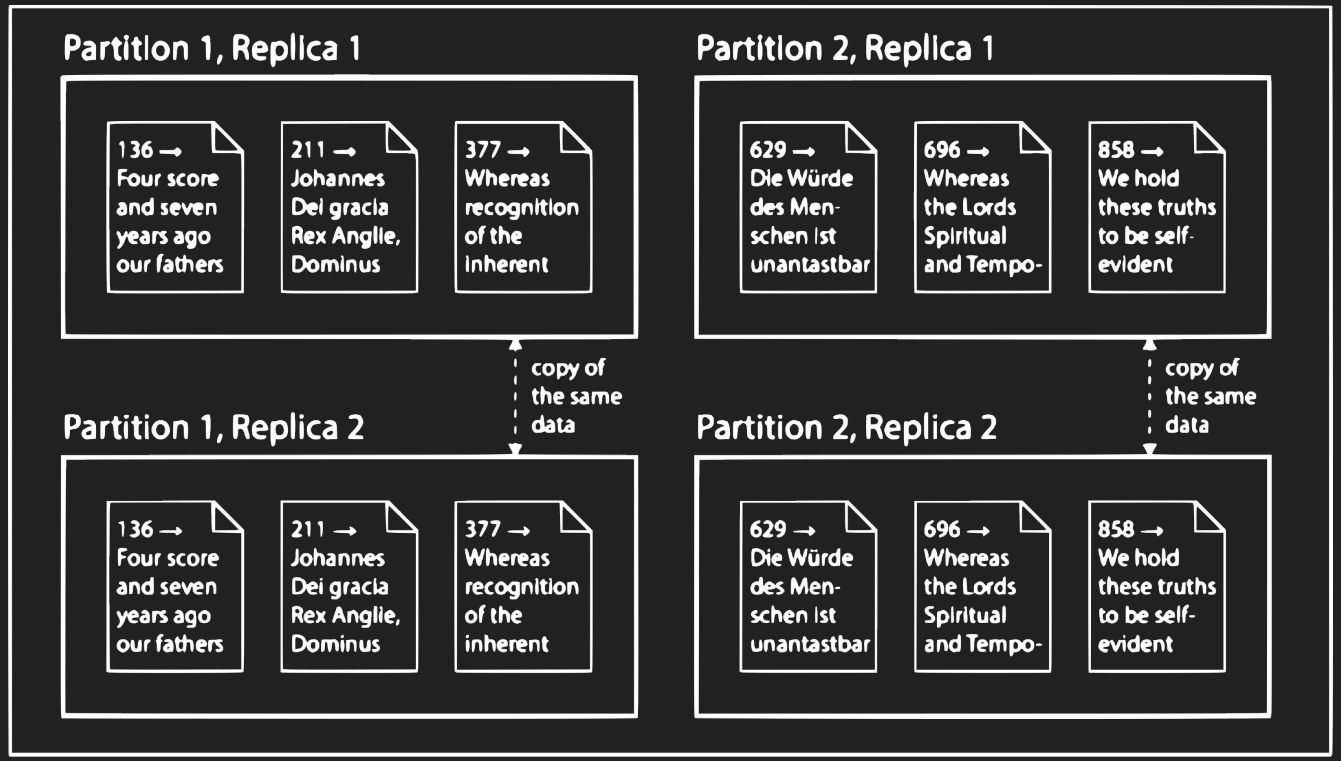


Figure II-1. A database split into two partitions, with two replicas per partition.

dataintensive.net
Designing Data-Intensive Applications
Martin Kleppmann

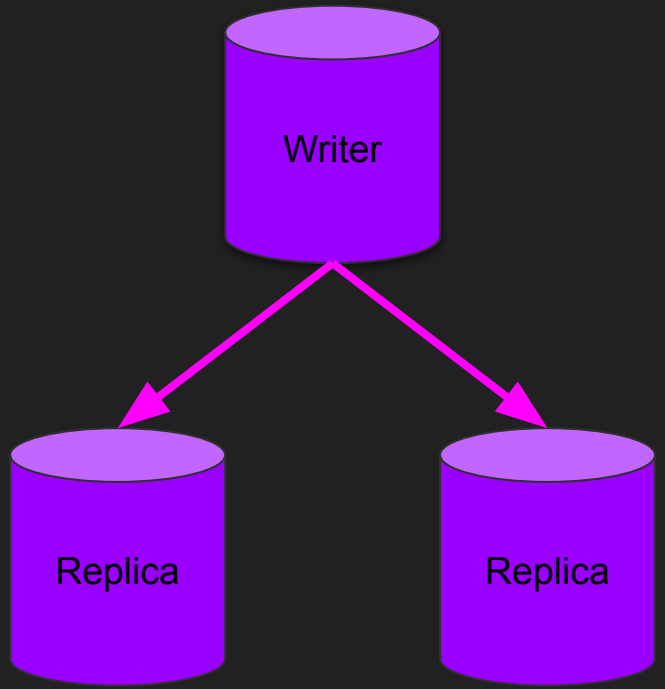
Datenbanken

So hier. Ihr kennt das. Datenbank. Bäm.

- Da gibt ja mehrere.
- Häufig dann auch viel Software.

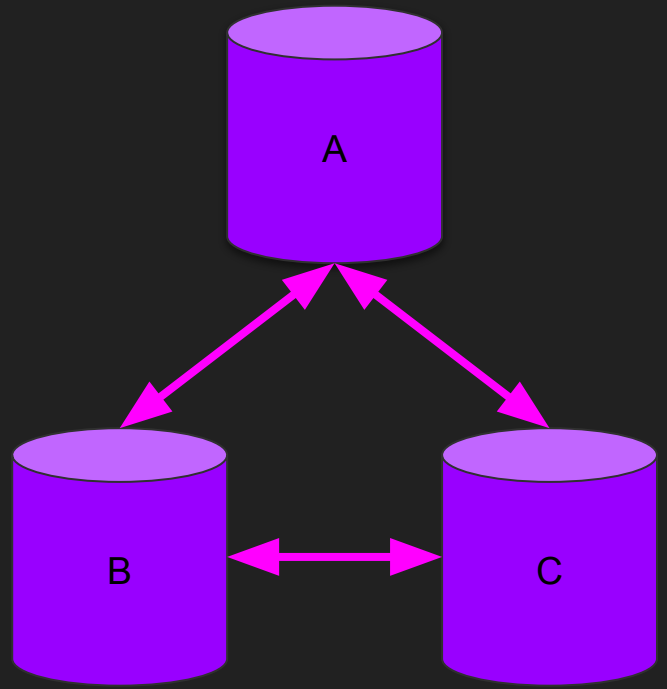
Datenbanken: Single Writer

- 1 Writer / Leader
- Viele Replicas / Follower
- Durchsatz beim Lesen
- Failover



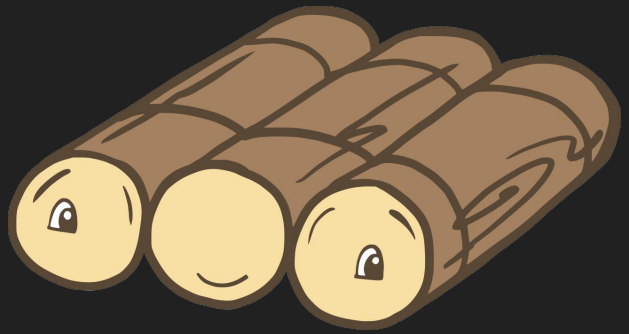
Datenbanken: Multi Writer

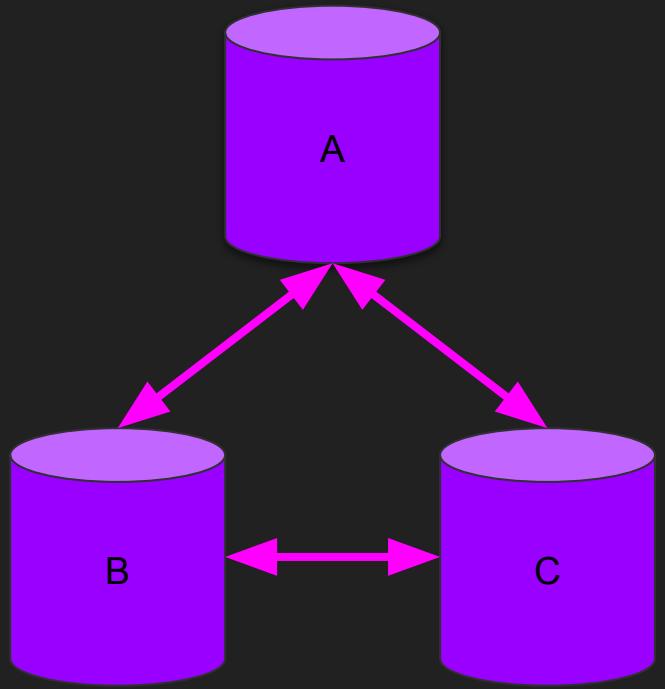
- Hier wird alles schlimmer 💥🍿
- Da geht doch was mit:
  - Menschen, wie bei git!
  - Quorum, vielleicht? 🤔
  - CRDTs 🤫



Datenbanken: Multi Writer Quorum

- Raft und Paxos für Datenbanken?
- w + r > n
- Einschränkungen?





Von Datenbanken zum Frontend

- Das Web als verteiltes System
- Local-First software
  - Privatsphäre
  - Verlässlichkeit
  - Unabhängiges Arbeiten
  - Kollaboration

Gemeinsames Tippen

- Alice und Bob editieren text ‘Hallo’
- Alice schreibt am Ende ‘!’
- Bob schreibt am Ende ‘ Nook’

Gemeinsames Tippen

- Alice und Bob editieren text ‘Hallo’
- Alice schreibt am Ende ‘!’
- Bob schreibt am Ende ‘ Nook’
- Carlos könnte sowas sehen:

t0:

- Hallo

t1:

- Hallo Nook
- Hallo!

t2:

- Hallo Nook!
- Hallo! Nook

Gemeinsames Tippen: Google Docs

- Operational Transformation
  - Concurrency control in groupware systems - C. Ellis, S. Gibbs, 1989

Gemeinsames Tippen: Google Docs

- Operational Transformation
  - Concurrency control in groupware systems - C. Ellis, S. Gibbs, 1989
- Server bekommt 2 Operations
  - (5, ‘ Nook’), (5, ’!’)
- Server etabliert Reihenfolge
  - (5, ‘ Nook’), (10, ‘!’)
- Alle sehen sowas:
  - Hallo Nook!

Gemeinsames Tippen: Google Docs

- Gibt es Einschränkungen?
  - Zentrale Server
  - ≤ 100 Schreibende

Gemeinsames Tippen: HedgeDoc

- CRDTs mit yjs
- Text CRDTs als Beispiel
- CRDTs: The Hard Parts
  - Martin Kleppmann, Hydra 2020

Text CRDTs mit Mengen

- Startwort ‘Halo’
- Das Interval (0, 1)

Text CRDTs mit Mengen

```
{position: 0.2, value: 'h'}
{position: 0.4, value: 'a'}
{position: 0.6, value: 'l'}
{position: 0.8, value: 'o'}
```

Text CRDTs mit Mengen

```
{position: 0.2, value: 'h'}
{position: 0.4, value: 'a'}
{position: 0.6, value: 'l'}
{position: 0.8, value: 'o'}
```

```
// Alice:
{position: 0.7, value: 'l'}

// Bob:
{position: 0.9, value: '!'}
```

Text CRDTs mit Mengen

```
{position: 0.2, value: 'h'}
{position: 0.4, value: 'a'}
{position: 0.6, value: 'l'}
{position: 0.7, value: 'l'}
{position: 0.8, value: 'o'}
{position: 0.9, value: '!'}
```

Text CRDTs mit Mengen

```
{position: 0.2, value: 'h'}
{position: 0.4, value: 'a'}
{position: 0.6, value: 'l'}
{position: 0.7, value: 'l'}
{position: 0.8, value: 'o'}
{position: 0.9, value: '!'}
```

```
// Alice:
{position: 0.82, value: 'N'}
{position: 0.84, value: 'o'}
{position: 0.86, value: 'o'}
{position: 0.88, value: 'k'}

// Bob:
{position: 0.82, value: 'W'}
{position: 0.84, value: 'e'}
{position: 0.86, value: 'l'}
{position: 0.88, value: 't'}
```

Text CRDTs mit Mengen

```
{position: 0.2, value: 'h'}
{position: 0.4, value: 'a'}
{position: 0.6, value: 'l'}
{position: 0.7, value: 'l'}
{position: 0.8, value: 'o'}
{position: 0.9, value: '!'}
```

halloNWeolokt!

```
// Alice:
{position: 0.82, value: 'N'}
{position: 0.84, value: 'o'}
{position: 0.86, value: 'o'}
{position: 0.88, value: 'k'}

// Bob:
{position: 0.82, value: 'W'}
{position: 0.84, value: 'e'}
{position: 0.86, value: 'l'}
{position: 0.88, value: 't'}
```

Text CRDTs mit Bäumen

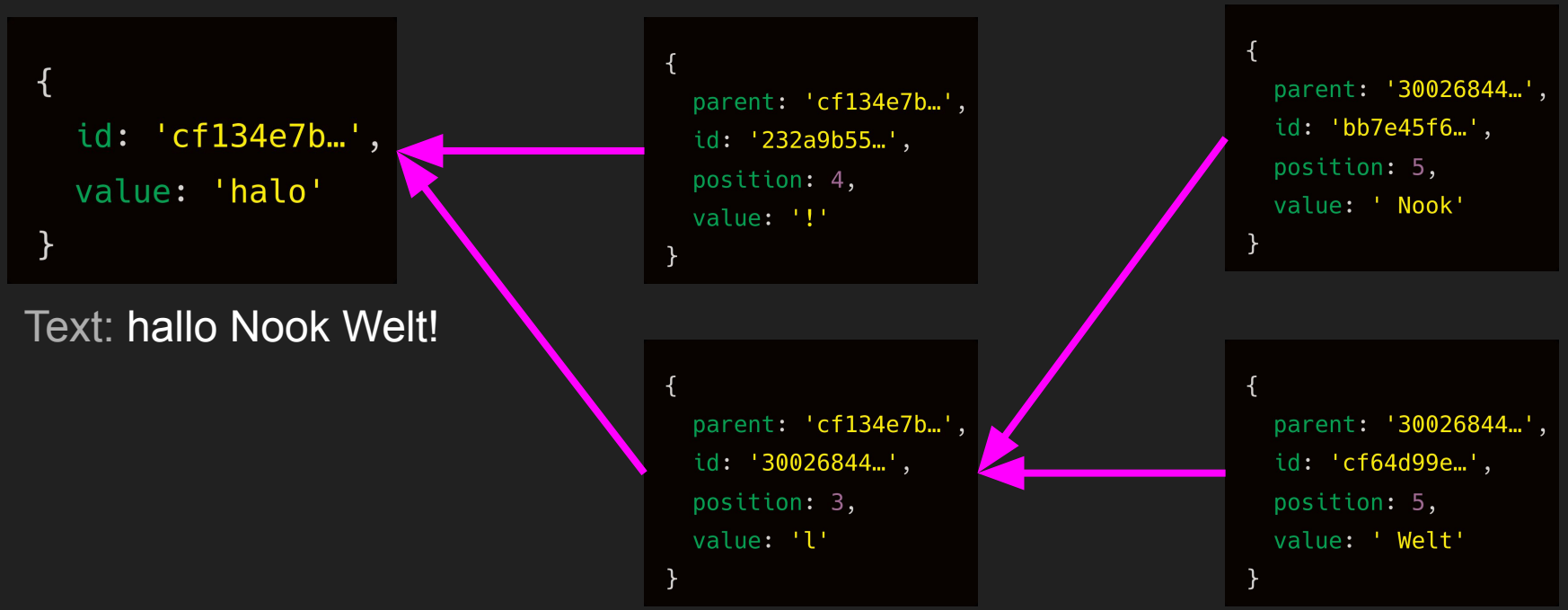
```
{
  id: 'cf134e7b…',
  value: 'halo'
}
```

Text: halo

Text CRDTs mit Bäumen

Text: hallo!

Text CRDTs mit Bäumen



Text: hallo Nook Welt!

CRDT laws 1/4

Konvergenz:

- Gleiche Änderungen?
- Gleicher Zustand!

CRDT laws 2/4

Idempotenz:

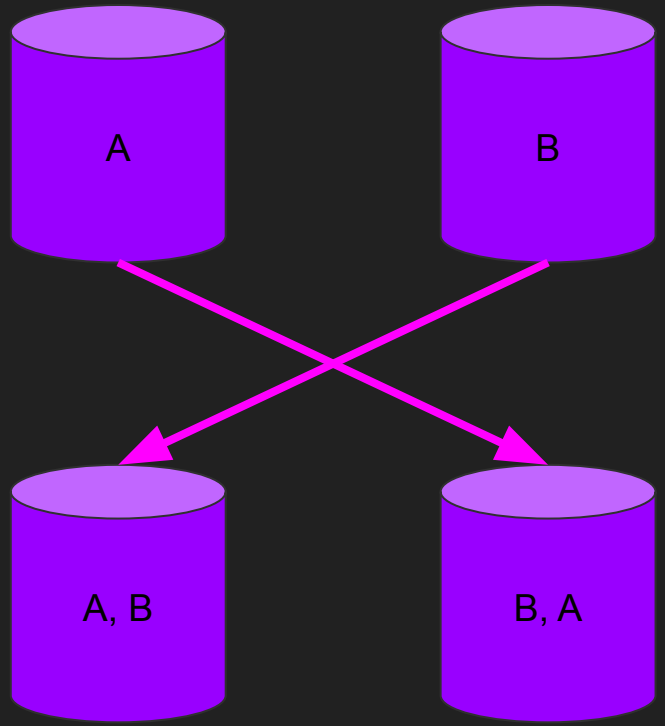
- merge(a, b) = merge(merge(a, b), b)

https://en.wikipedia.org/wiki/File:On_Off_-_Za%C5%82_Wy%C5%82_(3086204137).jpg

CRDT laws 3/4

Kommutativität:

- merge(a, b) = merge(b, a)



CRDT laws 4/4

Assoziativität:

- merge(a, merge(b, c)) = merge(merge(a, b), c)

CRDTs: Geschmacksrichtungen

- State-based
- Operation-based

CRDT Beispiele

- Counter
- Sets
- Registers
- Maps
- Append-only lists

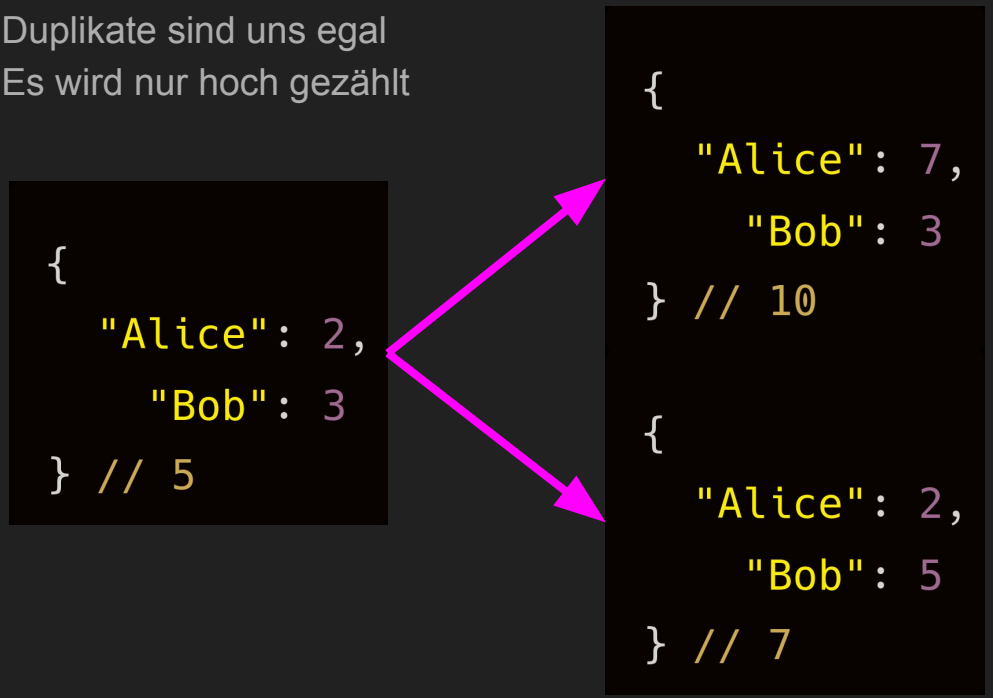
CRDT: grow-only counter

- Alice und Bob zählen Krähen
  - Duplikate sind uns egal
  - Es wird nur hoch gezählt

```
{
  "Alice": 2,
    "Bob": 3
} // 5
```

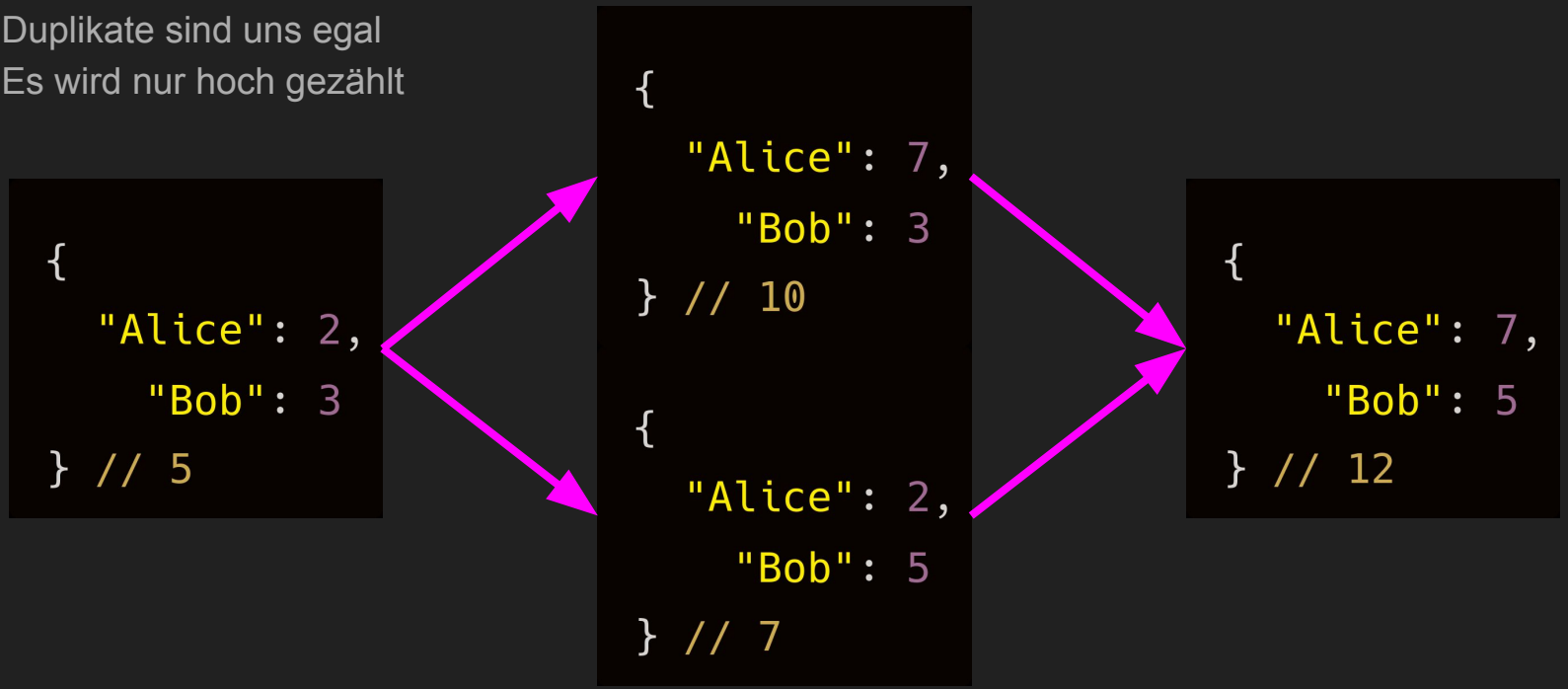
CRDT: grow-only counter

- Alice und Bob zählen Krähen
  - Duplikate sind uns egal
  - Es wird nur hoch gezählt



CRDT: grow-only counter

- Alice und Bob zählen Krähen
  - Duplikate sind uns egal
  - Es wird nur hoch gezählt



CRDT: Counter, state-based

- Alice und Bob zählen Krähen
  - Duplikate sind uns egal
  - Es wird nur hoch gezählt
- max(a, b) konvergiert
  - max(max(a, b), b) = max(a, b)
  - max(a, b) = max(b, a)
- Wenn wir aber runter zählen wollen?

```
{
  "Alice": 7,
    "Bob": 5
} // 12
```

CRDT: operation-based counter

- Alice und Bob zählen Krähen

```
{
  "id": "c67e40d2…"
  "value": 13
}
```

```
{
  "id": "6ad4553b…"
  "value": -5
}
```

CRDT: Grow-only sets

- Mengen von Dingen
- Immutability
- Zustände mergen mit $\cup$

CRDT: 2-phase sets

- Zwei grow-only sets
- Ein set für Dinge
- Ein set für Grabsteine

CRDT: 2-phase sets

- Zwei grow-only sets
- Ein set für Dinge
- Ein set für Grabsteine

Wir mergen so:

- $\cup$ auf Grabsteinen
- $\cup$ auf Dingen $\setminus$ $\cup$ auf Grabsteinen

CRDT: Register

- Veränderliche Werte
- Problem: Concurrency
- Multi-value register
- Last-write wins register

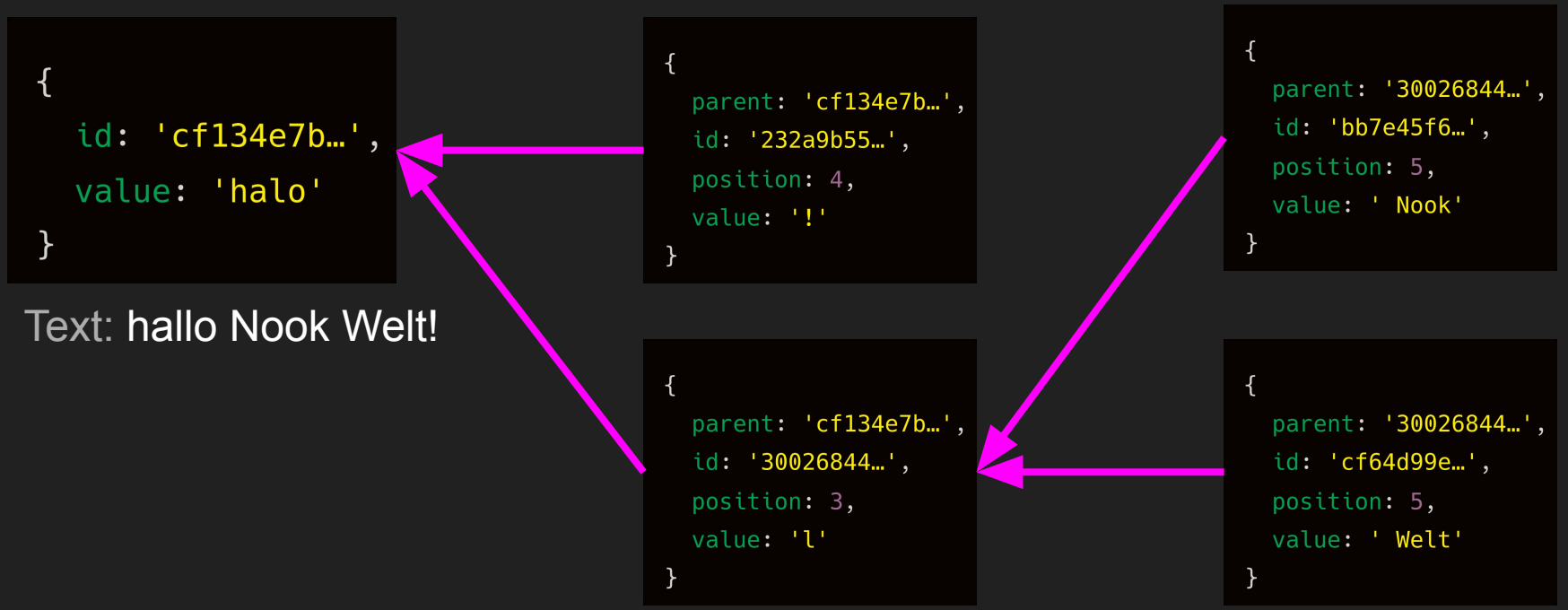
CRDT: Maps

- Sets von Tupeln
- Register für Kollisionen

```
type Map<K, V> = Set<{
  key: K,
  value: V,
  t: Timestamp
}>
```

CRDT: Append-only lists

- Wir erinnern uns an Text CRDTs mit Bäumen



Text: hallo Nook Welt!

Globale constraints

Was überhaupt los hier?

Globale constraints: Zählen

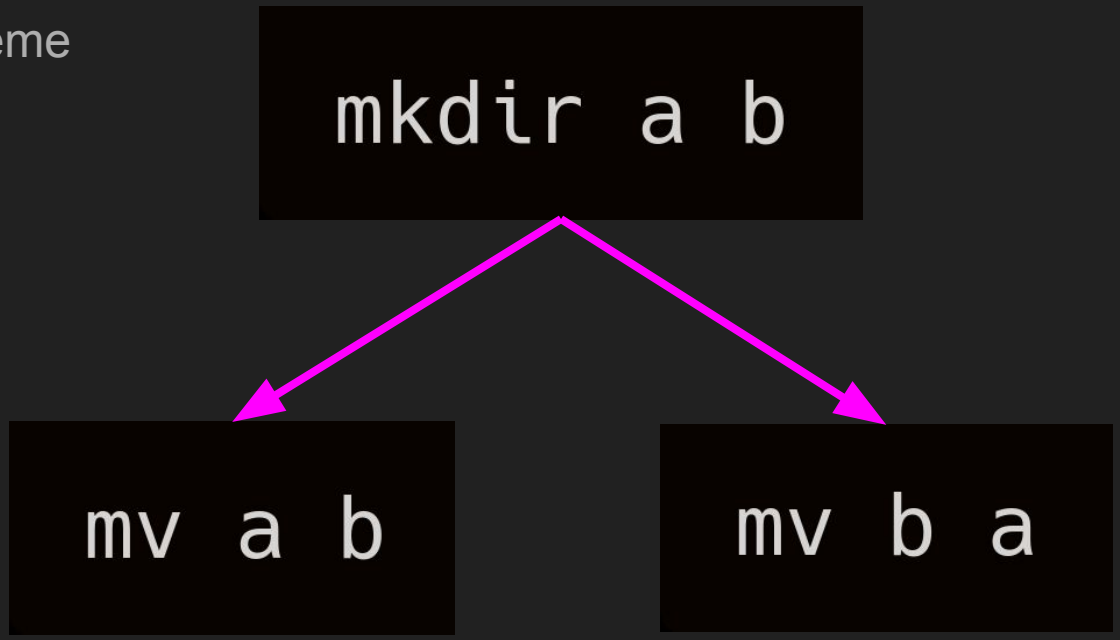
- Haben wir alle dabei?
- Zählen bis n

Globale constraints: Graphen

- Dateisysteme

Globale constraints: Graphen

- Dateisysteme



CRDTs: The Hard Parts

Conflict-free Replicated Data-Types (CRDTs)

Nook 2023

Jakob Runge

Links 1/4

- A comprehensive study of Convergent and Commutative Replicated Data Types
  - Marc Shapiro, Nuno Preguiça, Carlos Baquero, Marek Zawirski, 2011
- Designing Data-Intensive Applications
  - Martin Kleppmann, 2017
- End-to-End Arguments in System Design
  - J.H. Saltzer, D.P. Reed, D.D. Clark, 1984
- Local-First Software: You Own Your Data, in spite of the Cloud
  - Martin Kleppmann, Adam Wiggins, Peter van Hardenberg, Mark McGranaghan, 2019

Links 2/4

- Crdt.tech
- Raft.github.io
- Aphyr.com…jepsen-final-thoughts
- joelgustafson.com…merkelizing the key value store…
- cohost.org/tef/…how not to write a pipeline
- Programmingisterrible.com…how do you cut a monolith in half

Links 3/4 - YouTube

- CRDTs for mortals
  - James Long, dotJS 2019
- CRDTs for Non Academics
  - Russell Sullivan, 2017
- CRDTs and the Quest for distributed consistency
  - Martin Kleppmann, QCon 2018
- CRDTs: The Hard Parts
  - Martin Kleppmann, Hydra 2020

Links 4/4 - Wikipedia

- Conflict-free replicated data type
- CAP theorem
- Merkle Tree
- Distributed computing
- Fallacies of distributed computing
- Microservices - criticisms and concerns
- Paxos
- Operational Transformation