



Einführung in **CouchDB**

Zurücklehnen und entspannen!

<http://slog.io>

Thomas Schrader (@slogmen) 12/2010

Übersicht

- Bestandsaufnahme
- Ansatz
- Geschichte
- Technologien
- Features
- Skalierbarkeit
- Kurz & Gut
- Fazit

Relationale Datenbanken

- in den 70iger Jahren entwickelt
- für große Rechenanlagen
- komplexe Replikation
- unflexible, aufwendige Schemas
- SQL?

NoSQL

- *Not only SQL*
- nicht relationale Datenbanken
- Graph-Datenbanken
- Key-/Value-Datenbanken
- Tabellen (BigTable)
- **Dokumentenorientierte Datenbanken**

Ansatz

| Url | Title | Description |
|---|--------------|---|
| http://www.heise.de | Heise Online | Seite für IT-News |
| http://couchdb.org | CouchDB | Heimat der Dokumentenorientierten Datenbank CouchDB |
| http://fh-trier.de | FH Trier | Die Fachhochschule in Trier |

Ansatz

```
(  
  "Url": "http://couchdb.org"  
  "Title:" "CouchDB"  
  "Description: " "Heimat der ... CouchDB"  
)
```

Ansatz

- Schlüssel-/Wert Paare
- Verzicht auf Schemas: mehr Flexibilität, einfaches Design
- Passt zu Denkweise für Objekte in modernen Objektorientierten Sprachen

Geschichte

- frühe Doc-DB in Lotus Notes
- Damien Katz, Entwickler bei Lotus
- CouchDB wird seit 2005 entwickelt
- 2008-2009 Entwicklung von IBM gefördert
- 02/2008 Incubator der Apache Software Foundation
- 11/2008 vollwertiges Apache Projekt
- Apache Software License 2.0, *OpenSource!*

CouchDB

- Ziel: Dokumentenorientierten Ansatz aus Notes mit Map/Reduce-Algorithmen verbinden.
- Alles einfach halten!

Technologien

- Erlang
- JavaScript
- JSON
- RESTful API => HTTP!

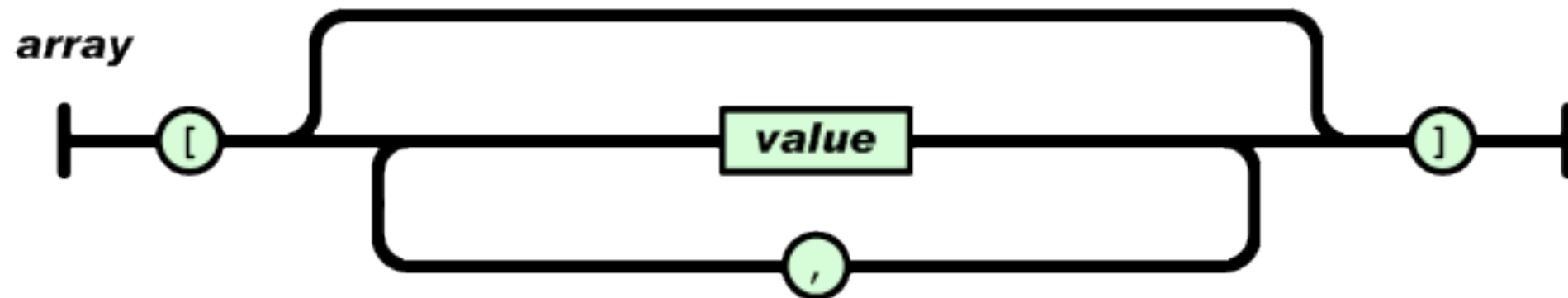
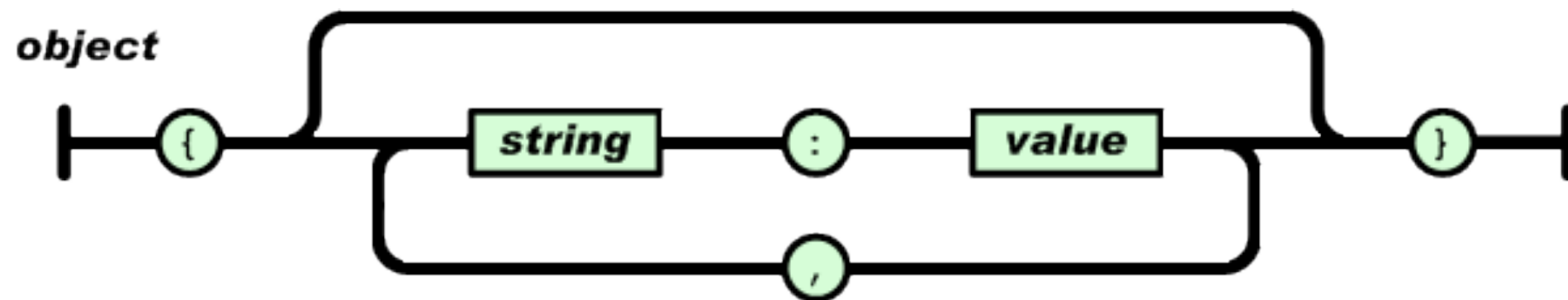
REST

- beschreibt eine Architektur
- **Representational State Transfer**
- Ressourcen, über URI identifizierbar
- einheitliche Schnittstelle

HTTP (CRUD)

- Create —→ POST /db/
- Retrieve —→ GET /db/docid
- Update —→ PUT /db/docid
- Delete —→ DELETE /db/docid

JSON



JSON

```
{  
  "_id": "BC4ea69ce1a73aa7d21d23b608d221d0",  
  "_rev": "1-967a00dff5e02add41819138abb3284d",  
  "type": "person",  
  "name": "Darth Vader",  
  "age": 63,  
  "headware": ["Helmet", "Sombrero"],  
  "dark_side": true  
}
```

API

- es werden keine speziellen APIs benötigt
- HTTP/JSON sind die API

API

```
slogbook:~ slogmen$ curl -X GET http://127.0.0.1:5984/_uuids
{"uuids":["e7f731bb2978e4e6871d0df45500299f"]}
slogbook:~ slogmen$ curl -X PUT http://127.0.0.1:5984/bookmarks/e7f731bb2978e4e6871d0df45500299f \
> -d '{"title":"CouchDB", "url":"http://couchdb.org"}'
{"ok":true,"id":"e7f731bb2978e4e6871d0df45500299f","rev":"1-6ce982d0c022b3457464787e3a5c4237"}
```


API

CouchDBX — Time to Relax

Overview > bookmarks > e7f731bb2978e4e6871d0df45500299f

✓ Save Document

⊕ Add Field

⬆ Upload Attachment...


✕ Delete Document...

Fields

Source

| Field | Value |
|---------|--------------------------------------|
| _id | "e7f731bb2978e4e6871d0df45500299f" |
| _rev | "1-6ce982d0c022b3457464787e3a5c4237" |
| ✕ title | "CouchDB" |
| ✕ url | "http://couchdb.org" |

← Previous Version | Next Version →



Co

Tools

Overview

Configuration

Replicator

Status

Test

Host

Recent

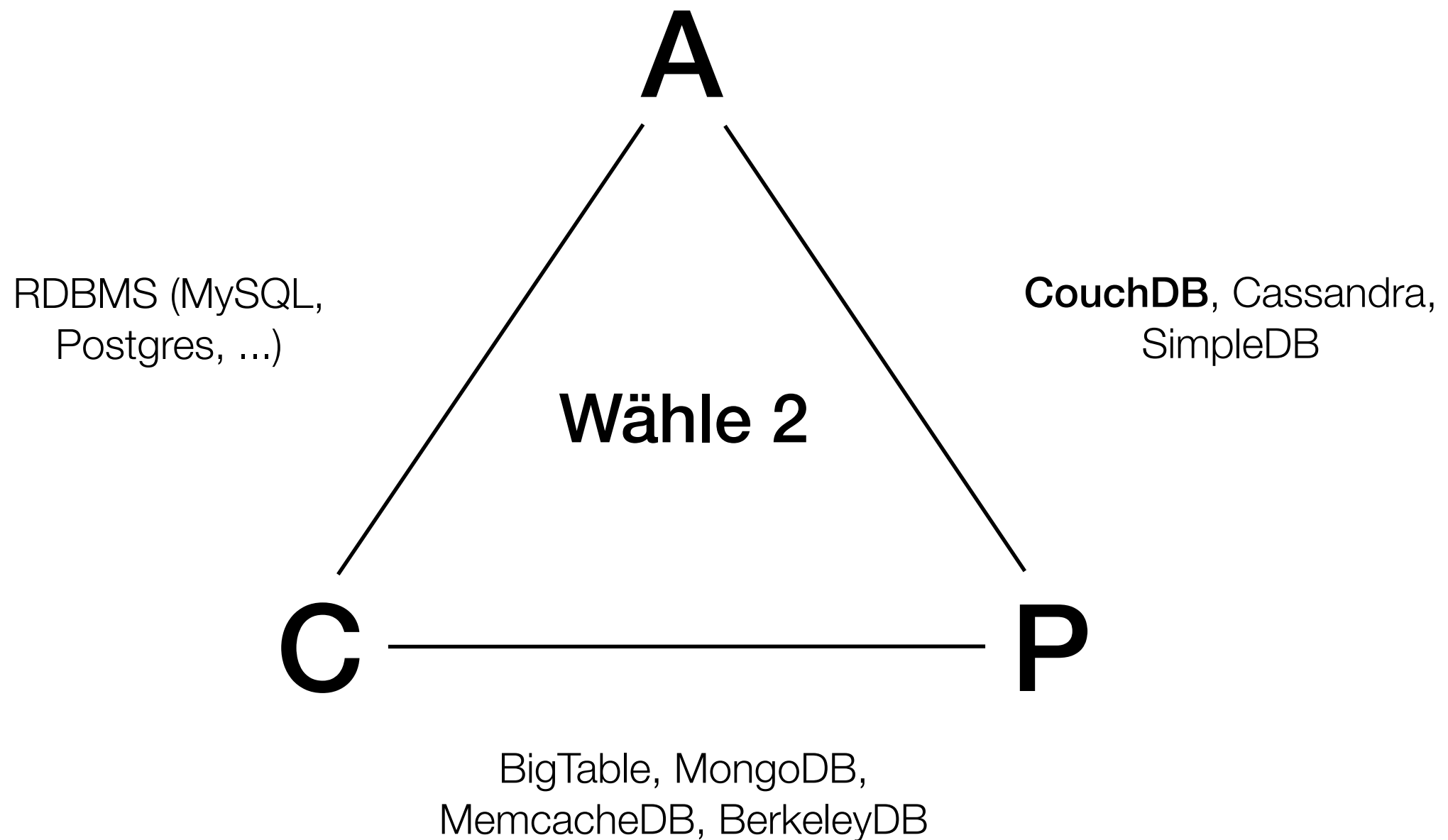
bookmarks

Future

lokale Konsistenz

- Dokumente werden versioniert:
 - **M**ulti **V**ersion **C**oncurrency **C**ontrol (MVCC)
 - Append Only
- kein Locking

verteilte Konsistenz



CAP Theorem

Eventual Consistency

- Verfügbarkeit steht über Konsistenz
- inkrementelle Replikation stellt Konsistenz wieder her

Features

- Views
- MapReduce
- Replikation

Views

- „Design-Dokumente“
- B+ Baum
- MapReduce Funktionen

MapReduce

```
{  
  "id": 1,  
  "date": 20101201,  
  "checkout": 100  
}
```

```
{  
  "id": 2,  
  "date": 20101201,  
  "checkout": 42  
}
```

```
{  
  "id": 3,  
  "date": 20101201,  
  "checkout": 23  
}
```

```
{  
  "id": 4,  
  "date": 20101201,  
  "checkout": 10  
}
```

MapReduce

100



```
{  
  "id": 1,  
  "date": 20101201,  
  "checkout": 100  
}
```

42



```
{  
  "id": 2,  
  "date": 20101201,  
  "checkout": 42  
}
```

23



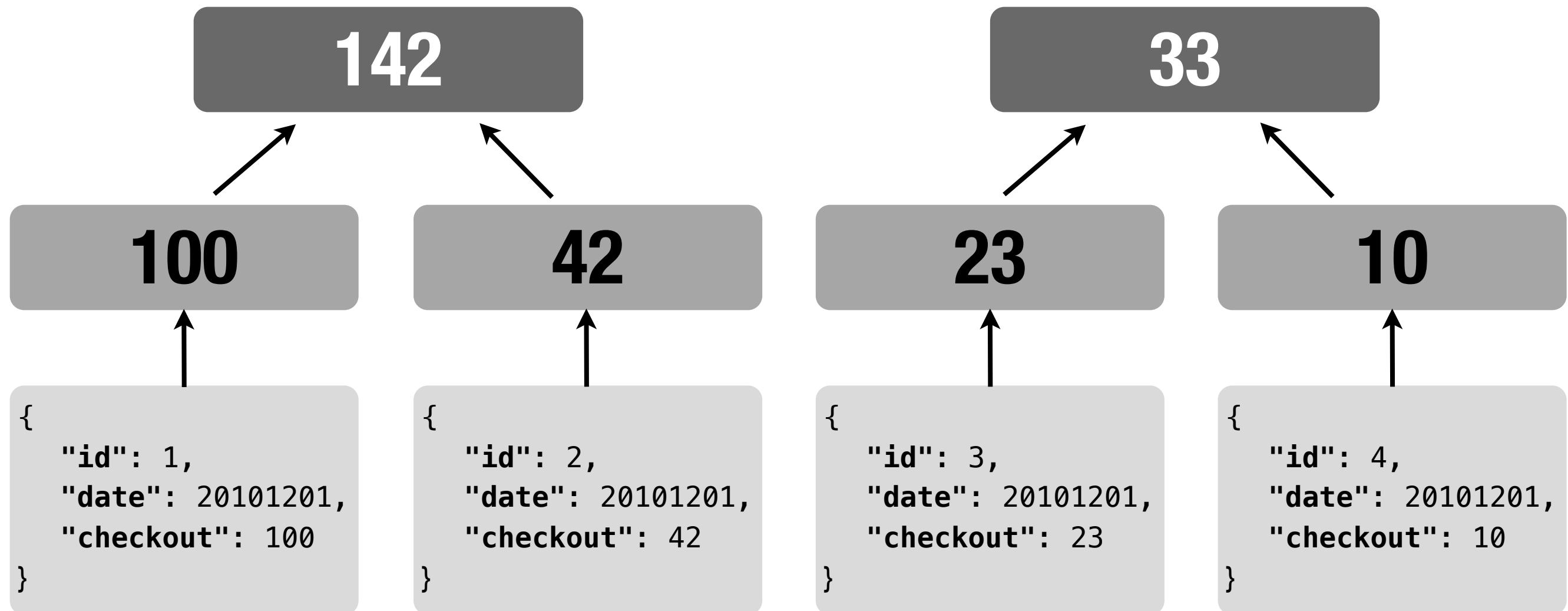
```
{  
  "id": 3,  
  "date": 20101201,  
  "checkout": 23  
}
```

10

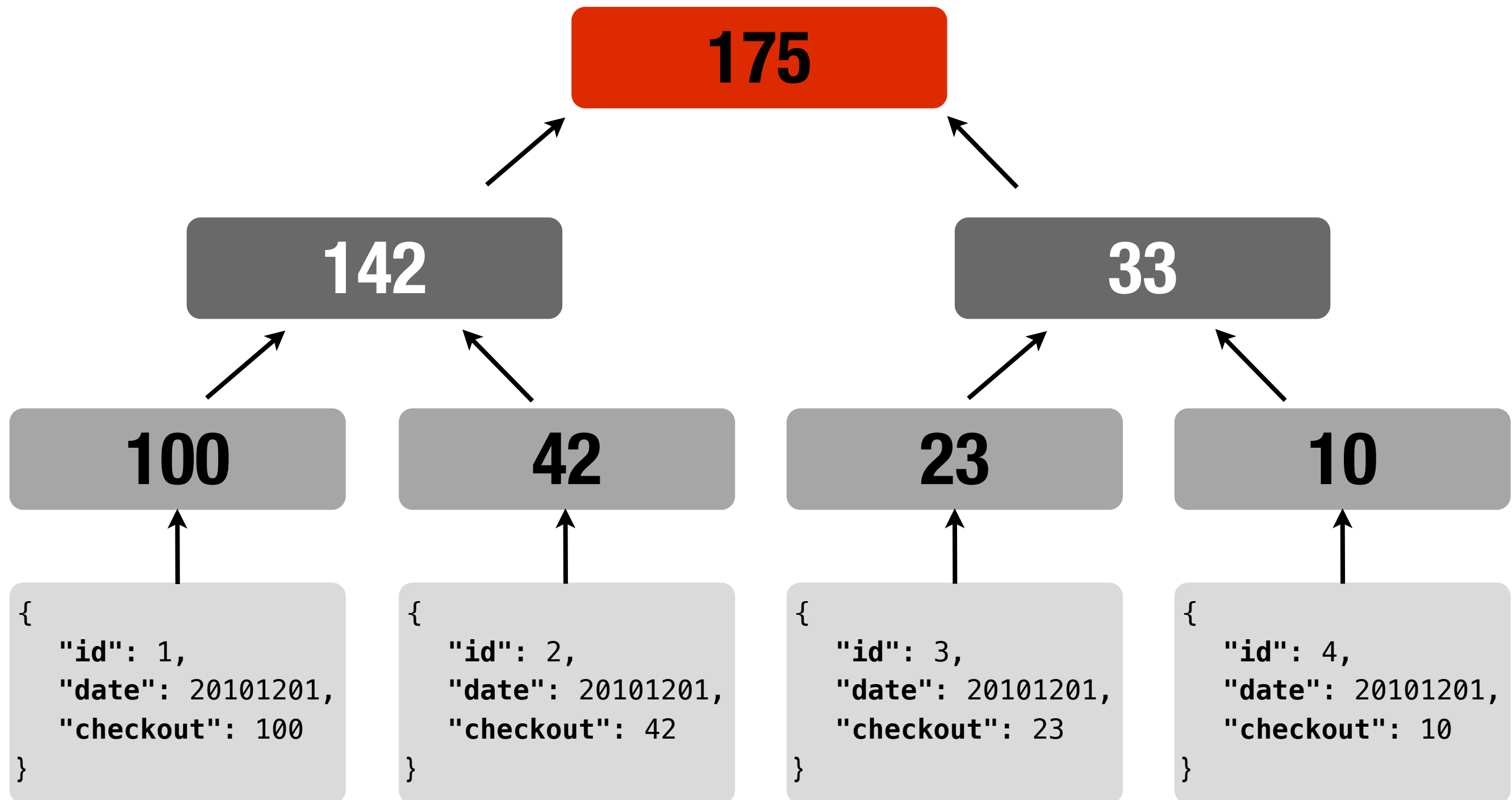


```
{  
  "id": 4,  
  "date": 20101201,  
  "checkout": 10  
}
```


MapReduce



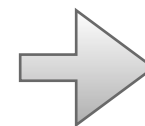
MapReduce

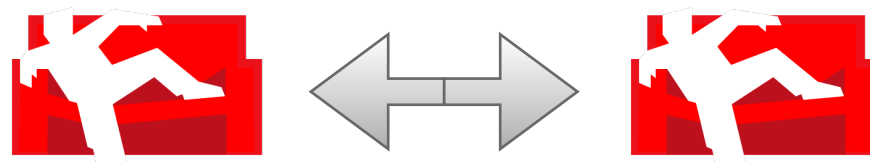


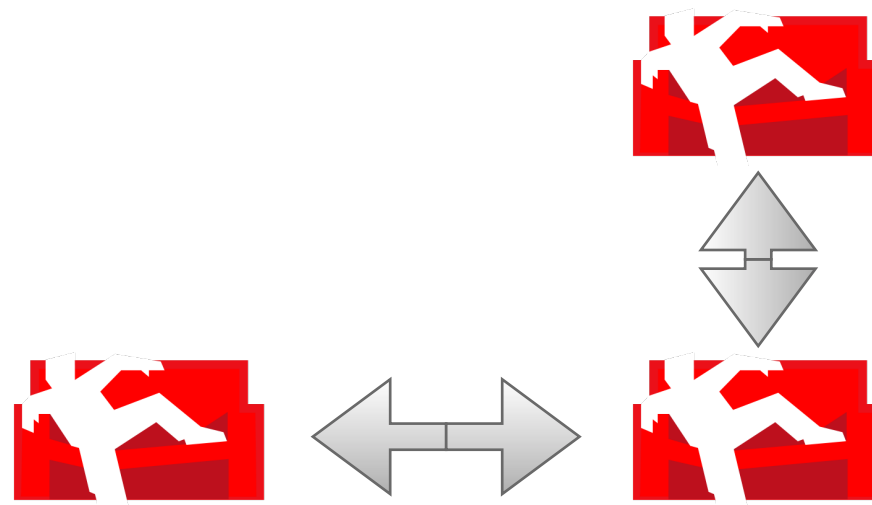
Replikation

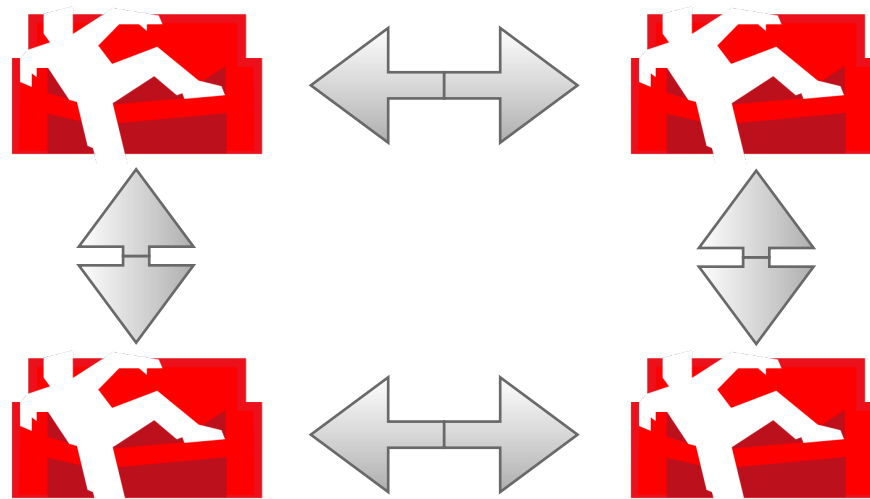
- Offline per default
- es gibt viele Realitäten
- Event-gesteuert
- Konfliktbehandlung

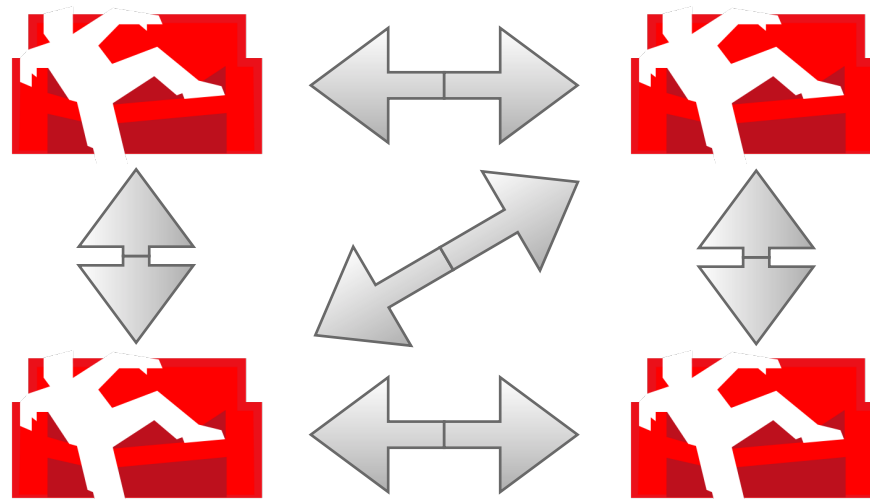


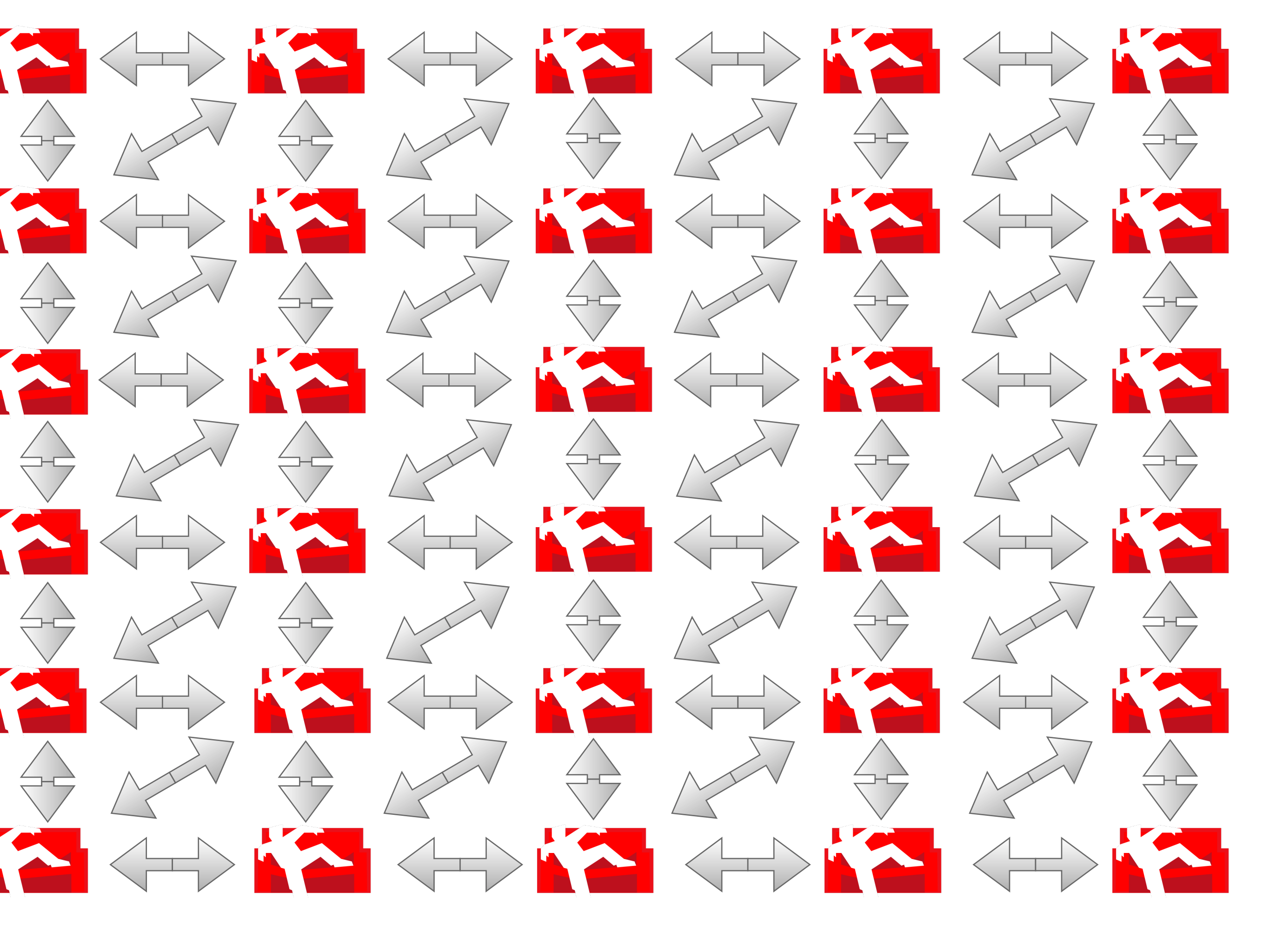












Konflikt-Behandlung

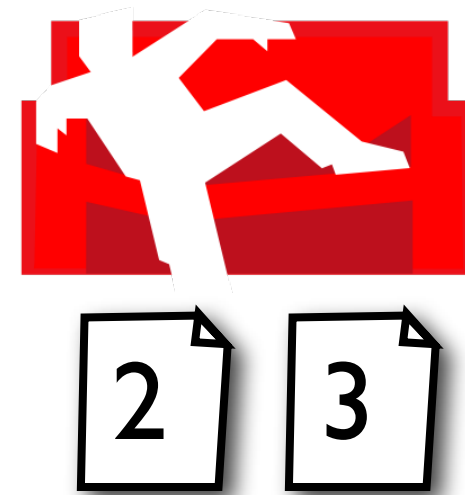
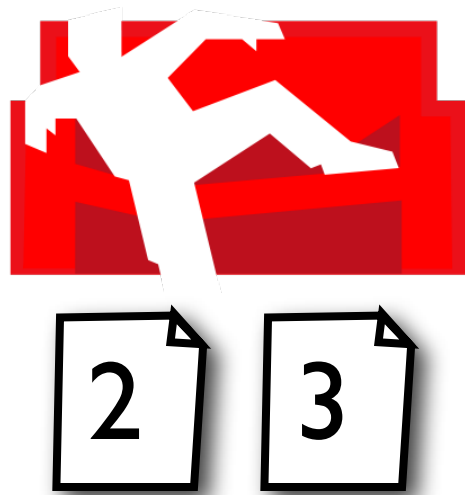


2

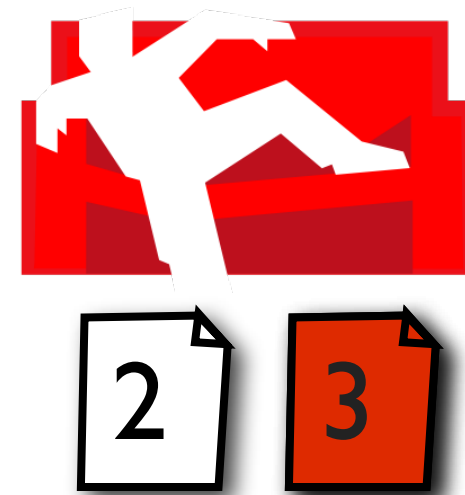


3

Konflikt-Behandlung



Konflikt-Behandlung



Skalierbarkeit

- horizontal:
 - Partitionierung (*Sharding*) mit CouchDB Lounge oder **BigCouch**
- vertikal:
 - Verteilen der Last auf viele Rechner
 - jemand eine Idee?

It's HTTP!

HTTP-Proxy-Server:
squid, Apache, nginx, ...

Kurz & Gut

Validatoren

CouchApps

Performance

Security

**CouchDB, die
Allzweckwaffe?**

naja, nicht ganz.

CouchDB

- in CouchDB geht es nicht um Relationen
- genügt nicht hohen Konsistenzanforderungen (ebay ...)

aber:

- Robust
- flexibel
- niedrige Lernkurve

Referenzen

- Ubuntu One
- BBC
- CERN

„Let me tell you something: Django may be built for the Web, but CouchDB is built of the Web. I’ve never seen software that so completely embraces the philosophies behind HTTP.“

Jacob Kaplan-Moss

<http://jacobian.org/writing/of-the-web/>

Fragen?

Time to Relax

Vielen Dank!

Quellen

- CouchDB, The Definitive Guide O'REILLY 2010
- „Umsetzung einer verteilten Anwendung mit der dokumentenorientierten Datenbank CouchDB“, Lena Herrmann Juli 2010
- Chaosradio Express 125
<http://chaosradio.ccc.de/cre125.html>
- Vortrag Jan Lehnardt NoSQL Berlin
- Vortrag Karl Glatz Oktober 2009
- Vortrag David Coallier, University Limerick 2010