

# CouchDB

Cluster of unreliable commodity hardware **D**ata **B**ase

- Was ist das,
- Wie funktioniert das und
- Warum macht Mensch das?





13. August 2008

## ■ Backronym

- Cluster
- of
- unreliable
- commodity
- hardware
- Data
- Base

## ■ Übersetzung

- Verbund
- aus
- unzuverlässiger
- Standard-
- Hardware
- Daten-
- bank

## ■ „Couch“, „Relax“

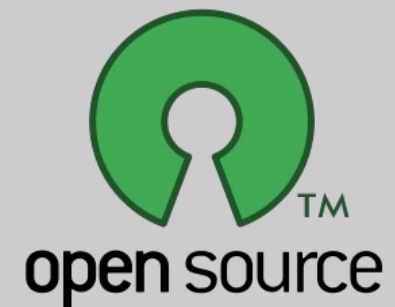
- Fokus auf Einfachheit
- Entspannung bei der Entwicklung und Benutzung

# CouchDB – Was ist das?



13. August 2008

- Dokumentenorientierte Datenbank
  - **Nicht** relational wie die meisten populären Datenbanken (MySQL, PostgreSQL, Firebird, DB2, MSSQL, Oracle)
  - Eher wie Lotus Notes
- Open Source Software
  - Seit Ende 2008 „Apache Project“
  - 7 Entwickler („Committer“)
    - Damien Katz
    - Jan Lehnardt
- Einfachheit steht im Vordergrund
  - Gall's Law

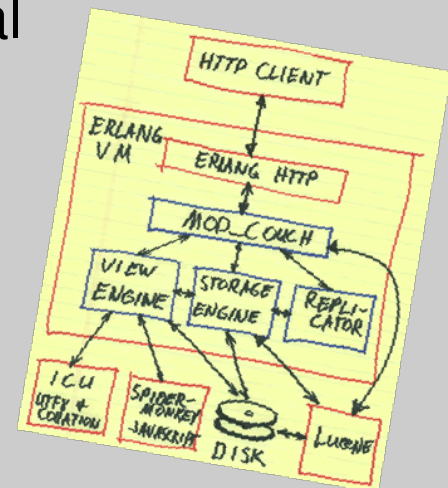


# CouchDB – Wie funktioniert das?



13. August 2008

- Entwickelt in Erlang
- Zwischen Client und Server
  - JSON-Objekte zum Datenaustausch
  - RESTful HTTP zur Kommunikation
- Auf dem Server selbst
  - Zugriff ausschließlich über Indizes (oder UUID)
    - Werden mittels JavaScript zur Abfragezeit einmal aufgebaut
  - Arbeitet mit atomaren Operationen
    - Erlaubt gleichzeitige Reads und Writes

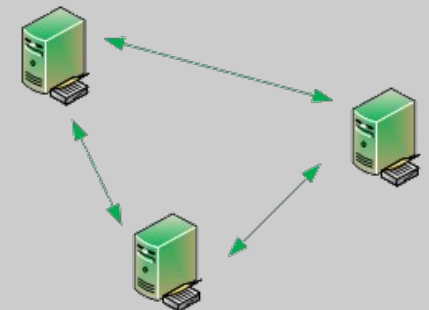


# CouchDB – Die Vorteile



13. August 2008

- Einfach zu bedienende Schnittstelle
- Flexible und simple Datenstruktur
  - Perfekt für die meisten Web 2.0-Anwendungen
  - Skriptbare Indizes (!!!)
- Gute Replikations-Features
  - Von Haus aus nur manuell (was zum basteln!)
  - Bi-Direktionales Konflikt-Management
  - Wie Lotus Notes gut für („disconnected“) Offline-Operationen geeignet
  - „The Eight Fallacies of Distributed Computing“



# Fertig!



13. August 2008

- ✓ Vielen Dank für eure Aufmerksamkeit
  
- Fragen?
- Wünsche?
- Anregungen?
  
- Webseite: [www.couchdb.org](http://www.couchdb.org)
- Kommerzielles Angebot: [www.couch.io](http://www.couch.io)
  
- E-Mail: [bernd@tiggerswelt.net](mailto:bernd@tiggerswelt.net)



13. August 2008

## The Eight Fallacies of Distributed Computing

*Peter Deutsch*

Essentially everyone, when they first build a distributed application, makes the following eight assumptions. All prove to be false in the long run and all cause *big* trouble and *painful* learning experiences.

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

For more details, read the article by Arnon Rotem-Gal-Oz

Quelle: <http://blogs.sun.com/jag/ressource/Fallacies.html>