

Prof. DI Dr. Erich Gams

Einführung und Anwendung

MongoDB

informationssysteme htl-wels

Übersicht ➡ Was lernen wir?



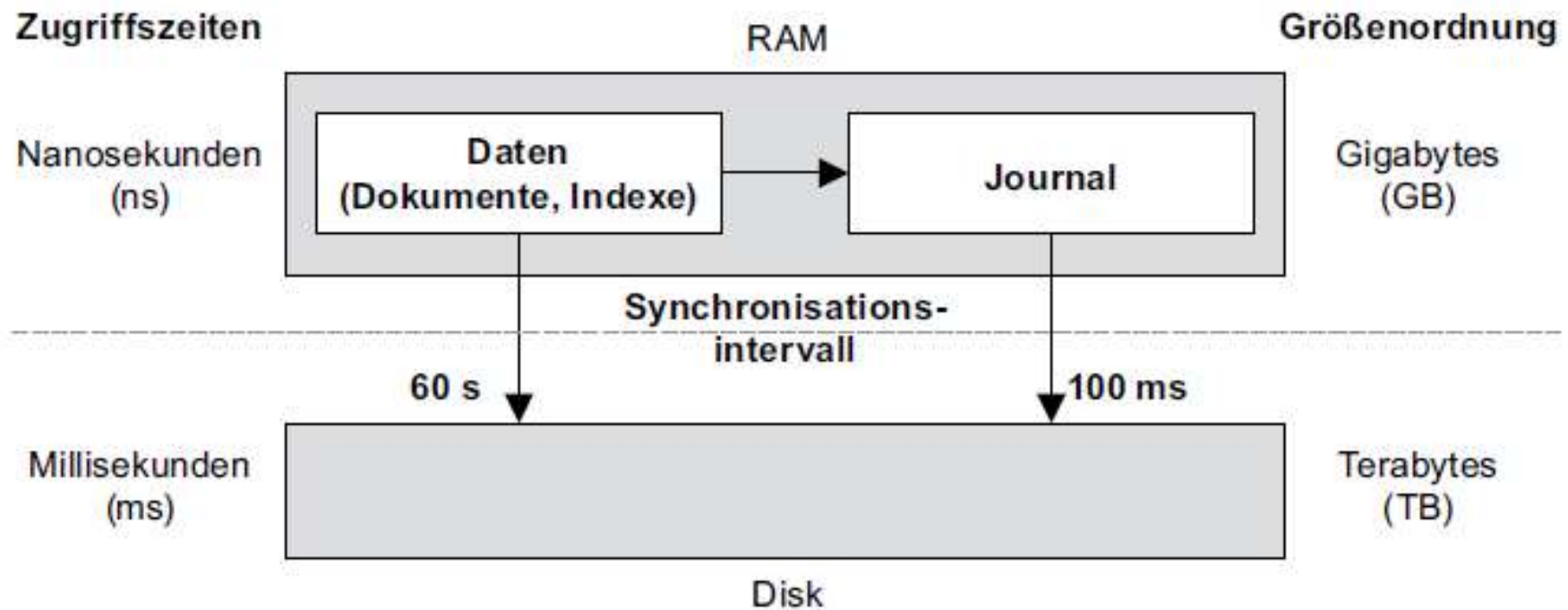
- › Internas
- › Replikation
- › Hands-on -> Tutorial



Interne Arbeitsweise eines MongoDBServers

- › Muss das System mit großen Datenmengen effizient umgehen können?
- › Werden hohe Anforderungen an Skalierbarkeit und Flexibilität der Datenbank gestellt?
- › Sollen sich die Daten über mehrere Server verteilen lassen?
- › Sind häufige Änderungen an der Datenstruktur in Zukunft zu erwarten?

Interne Arbeitsweise eines MongoDBServers



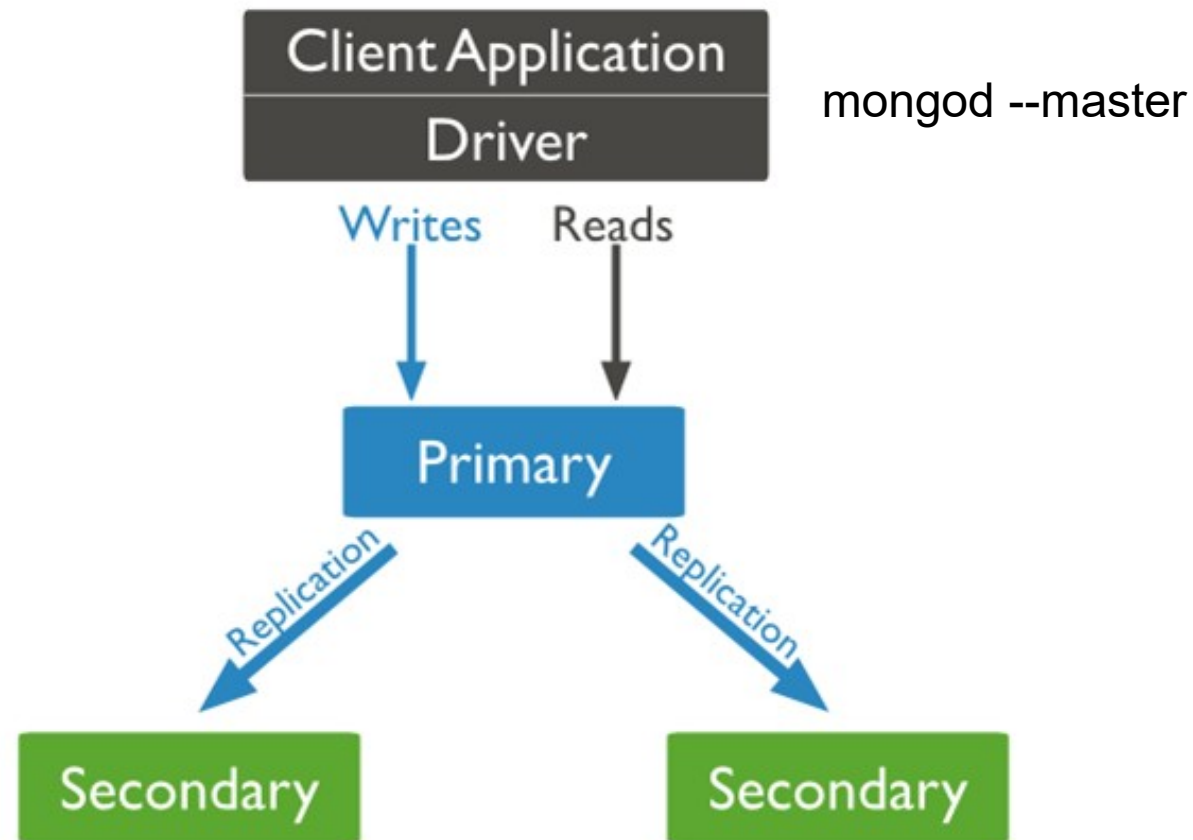
Interne Arbeitsweise eines MongoDBServers

- › Daten werden zunächst im RAM abgelegt und dann auf Disk synchronisiert
 - Geschwindigkeit: (Nano- versus Millisekunden).
 - Defaultwert: 60 Sekunden
 - Befehl: `--syncdelay <Sekunden>`
- › Zur Vermeidung von Datenverlust
 - Protokoll: Journaling
 - Synchronisation: alle 100 Millisekunden
 - Befehl: `--journalCommitInterval <milliseconds>`

Replikation

- › “Replication provides redundancy and increases data availability”
- › Grundlage: Oplog (Operations log)
 - Realisiert als Ringpuffer (Capped Collection)
 - Alle schreibenden Operationen werden festgehalten
 - *local.oplog.rs*

Master Slave Replication („deprecated“)



Mongod - - slave - -source <master-host:port>

Replica Sets

- › Flexible Rollenverteilung
- › Ein Primary Knoten (über den die Schreibzugriffe ausgeführt werden)
- › Jeder Knoten muss fähig sein, die Maximallast zu tragen
- › Ein MongoDB Treiber handelt aus, welcher Server gerade der Primary ist.

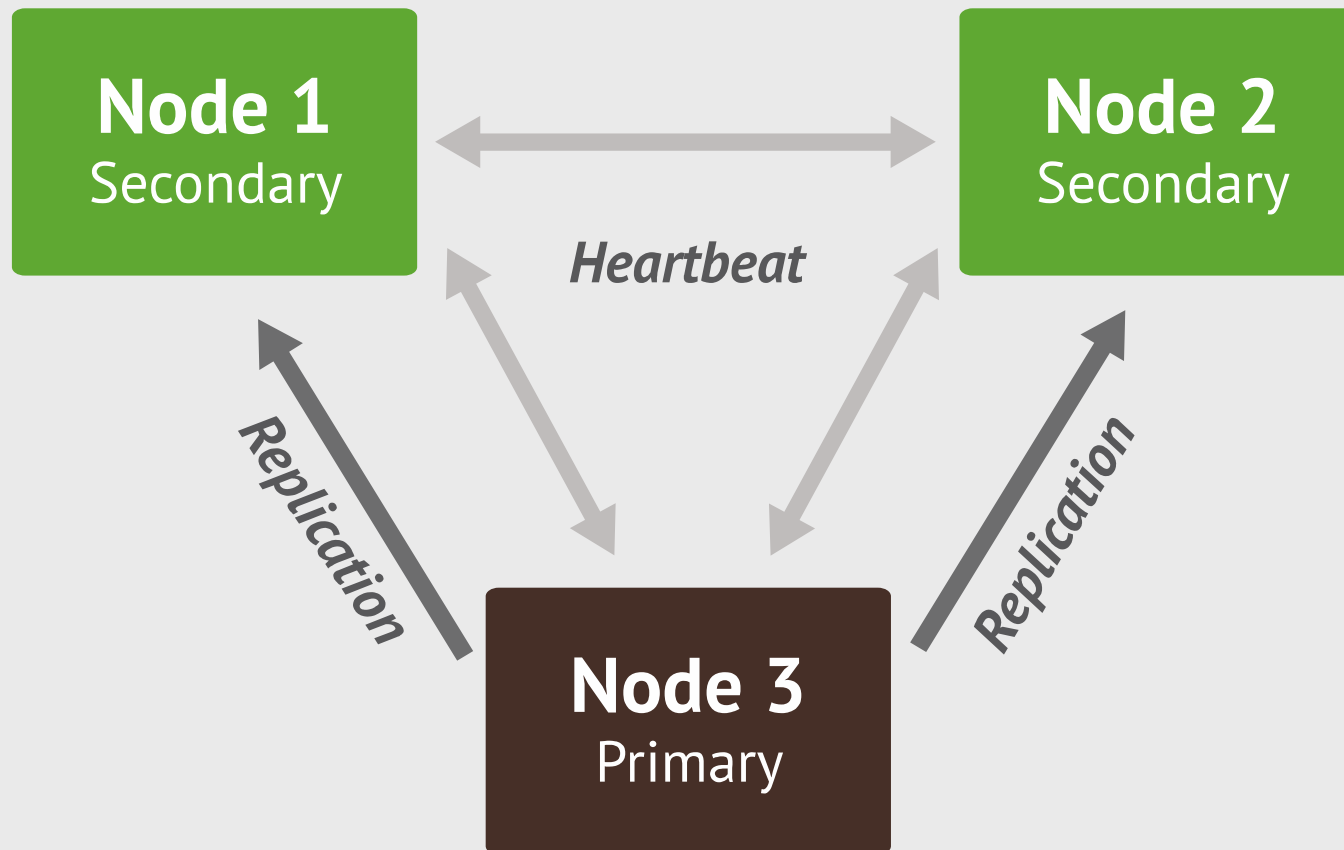
Replica Set – Creation

Node 1

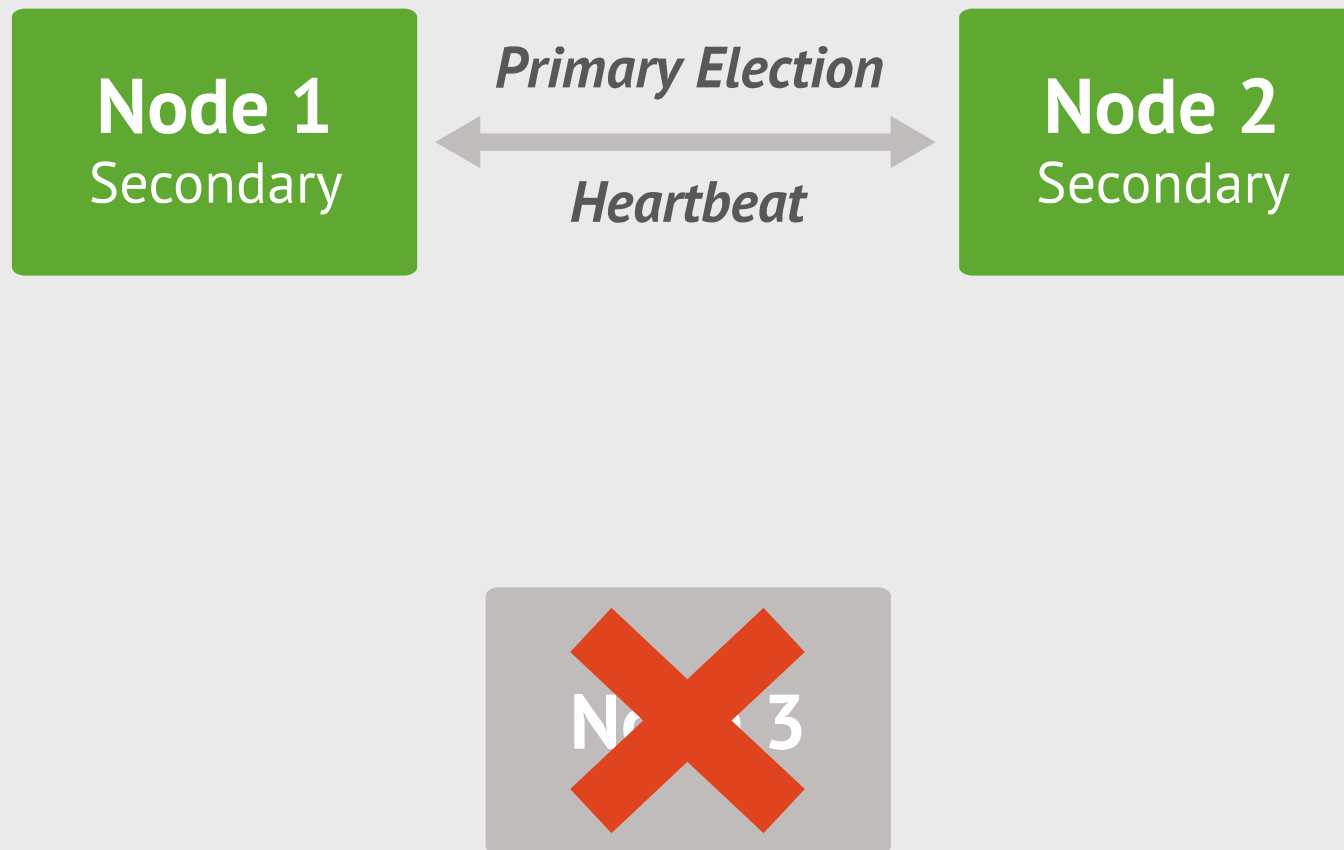
Node 2

Node 3

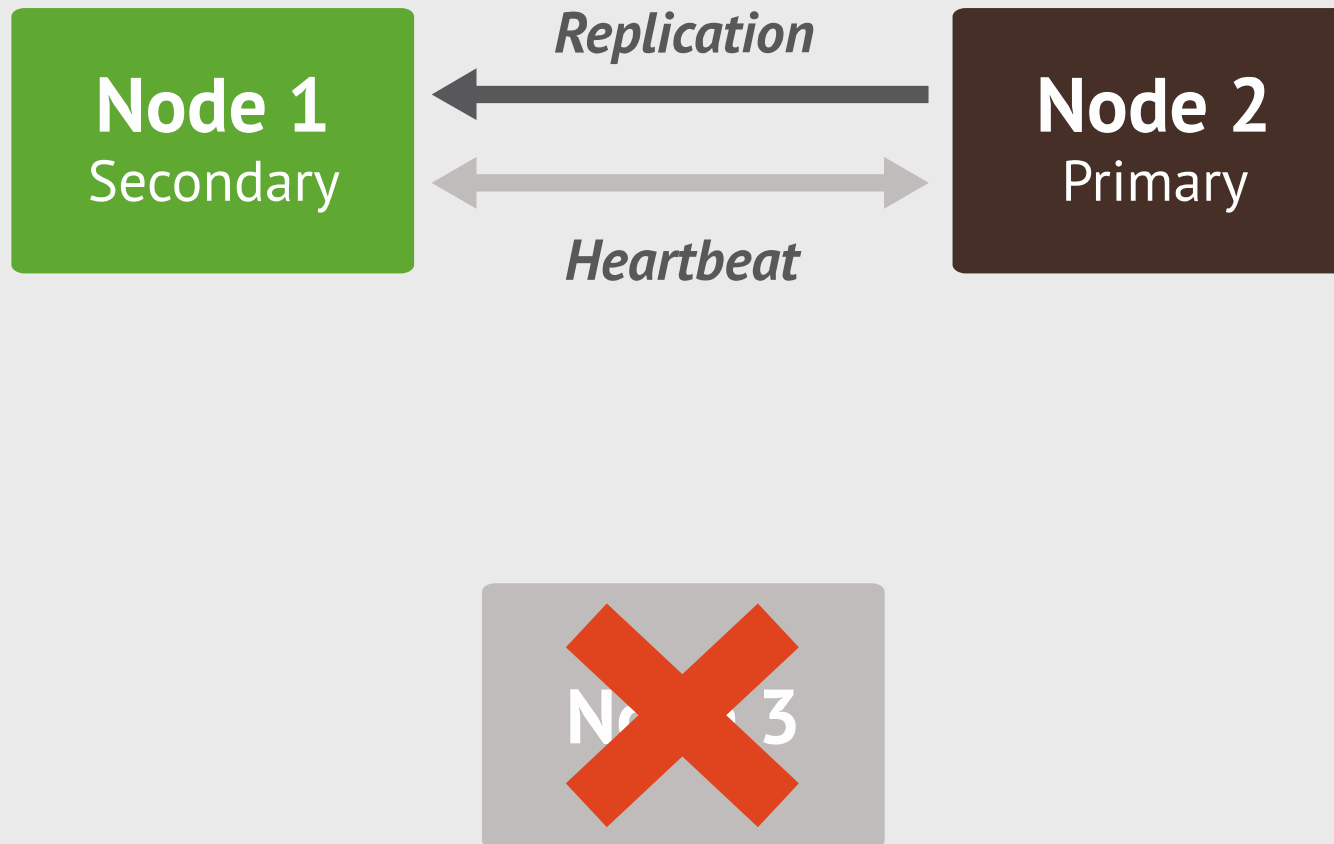
Replica Set – Initialize



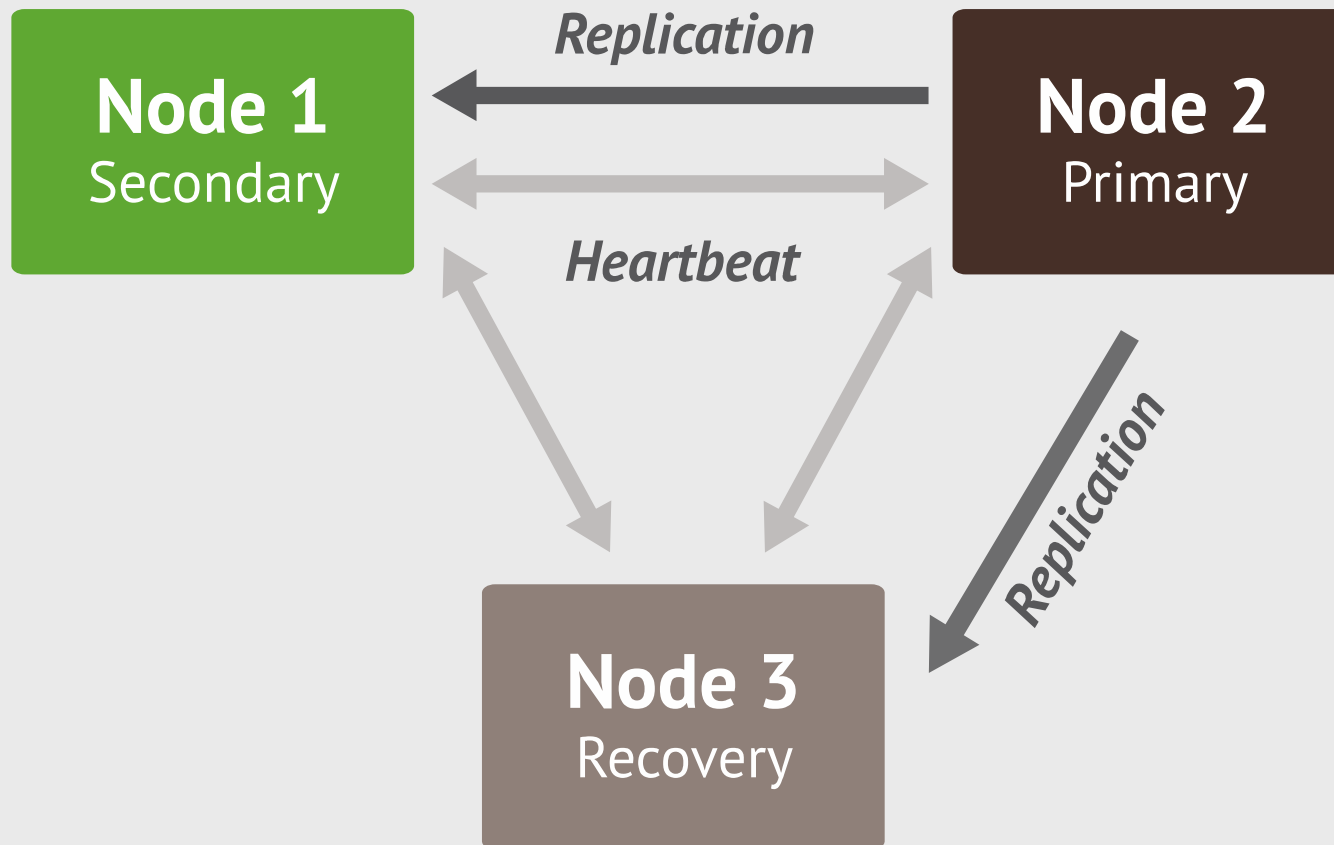
Replica Set – Failure



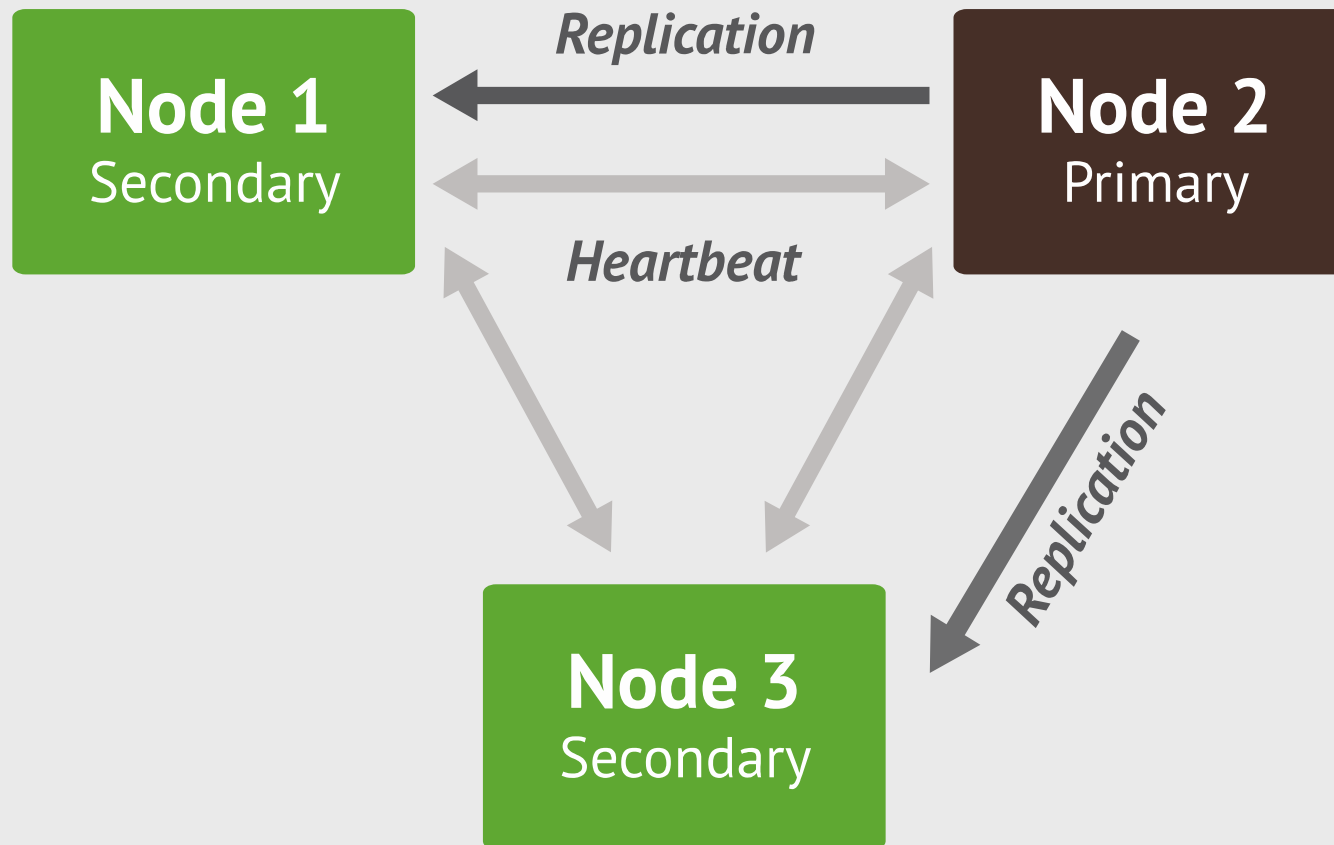
Replica Set – Failover



Replica Set – Recovery



Replica Set – Recovered



Arbiter

- › An arbiter does **not** have a copy of data set and **cannot** become a primary.
- › Replica sets may have arbiters to add a vote in *elections of for primary*.
- › Arbiters *always* have exactly 1 vote election, and thus allow replica sets to have an uneven number of members, without the overhead of a member that replicates data.
- › (<https://docs.mongodb.org/manual/core/replica-set-arbiter/>)

Write Concern

- › Wieviele Mitglieder eines Replica Sets müssen die Schreiboperation bestätigen müssen (bevor sie als erfolgreich gilt).
- › Explizite Angabe der Knoten
- › „majority“ – die Mehrheit der verfügbaren Knoten
- › Sicherheit vs Performance
 - Je weniger desto schneller

Read Preferences

- › „Strategie für lesenden Zugriff“
- › „Primary“ oder „primaryPreferred“
 - Primary bevorzugt
 - Konsistenz bevorzugt
- › „secondary“ oder „secondaryPreferred“
 - Horizontale Skalierung
- › Bei geografischer Verteilung
 - „nearest“ – gemessen an Ping-Zeit

Replica Sets

› Übung!

Auf los geht's los ;-)

