



# **MySQL High Availability**

## **DOAG 2013 Datenbank**

### **14. Mai 2013, Düsseldorf**

## **Oli Sennhauser**

**Senior MySQL Berater, FromDual GmbH**

**[oli.sennhauser@fromdual.com](mailto:oli.sennhauser@fromdual.com)**

# Über FromDual GmbH

- **FromDual bietet neutral und unabhängig:**
  - **Beratung für MySQL**
  - **Support für MySQL und Galera Cluster**
  - **Remote-DBA Dienstleistungen für MySQL**
  - **MySQL Schulungen**
- **Oracle Silver Partner (OPN)**



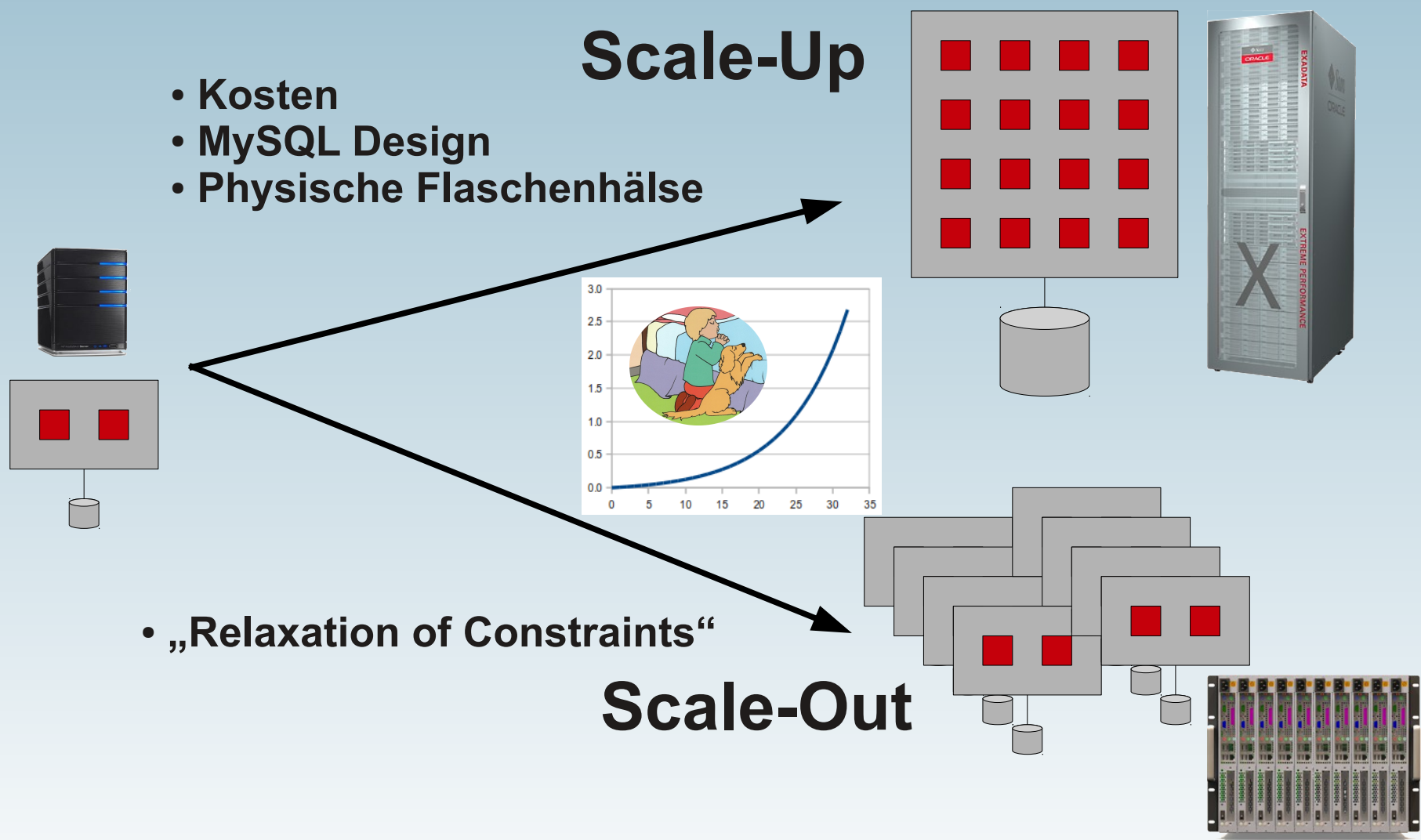
[www.fromdual.com](http://www.fromdual.com)

# Inhalt

## MySQL High Availability

- › **Scale-Out vs. Scale-Up**
- › **Master/Slave Replikation**
- › **Master/Master Replikation**
- › **Aktiv/passiv failover Cluster mit SAN**
- › **Aktiv/passiv failover Cluster mit DRBD**
- › **Galera Cluster für MySQL**
- › **MySQL (NDB) Cluster**

# MySQL Scale-Out vs Scale-Up [www.fromdual.com](http://www.fromdual.com)



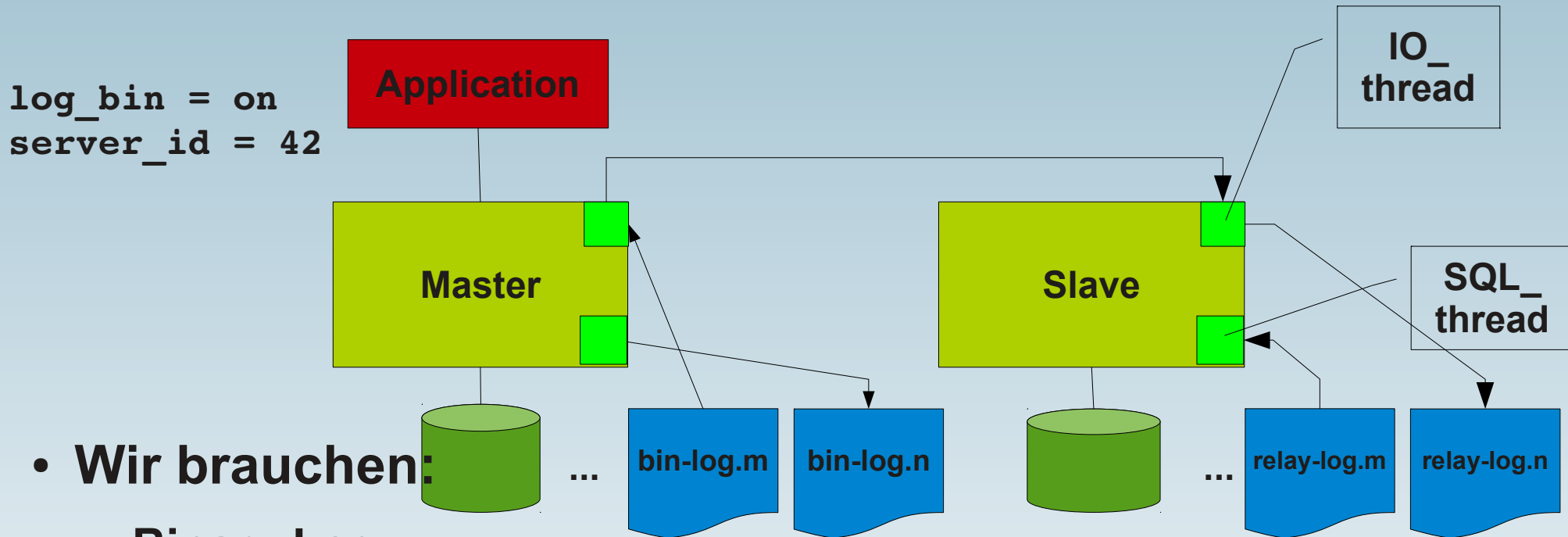
- Kosten
- MySQL Design
- Physische Flaschenhalse

- „Relaxation of Constraints“

**Scale-Up**

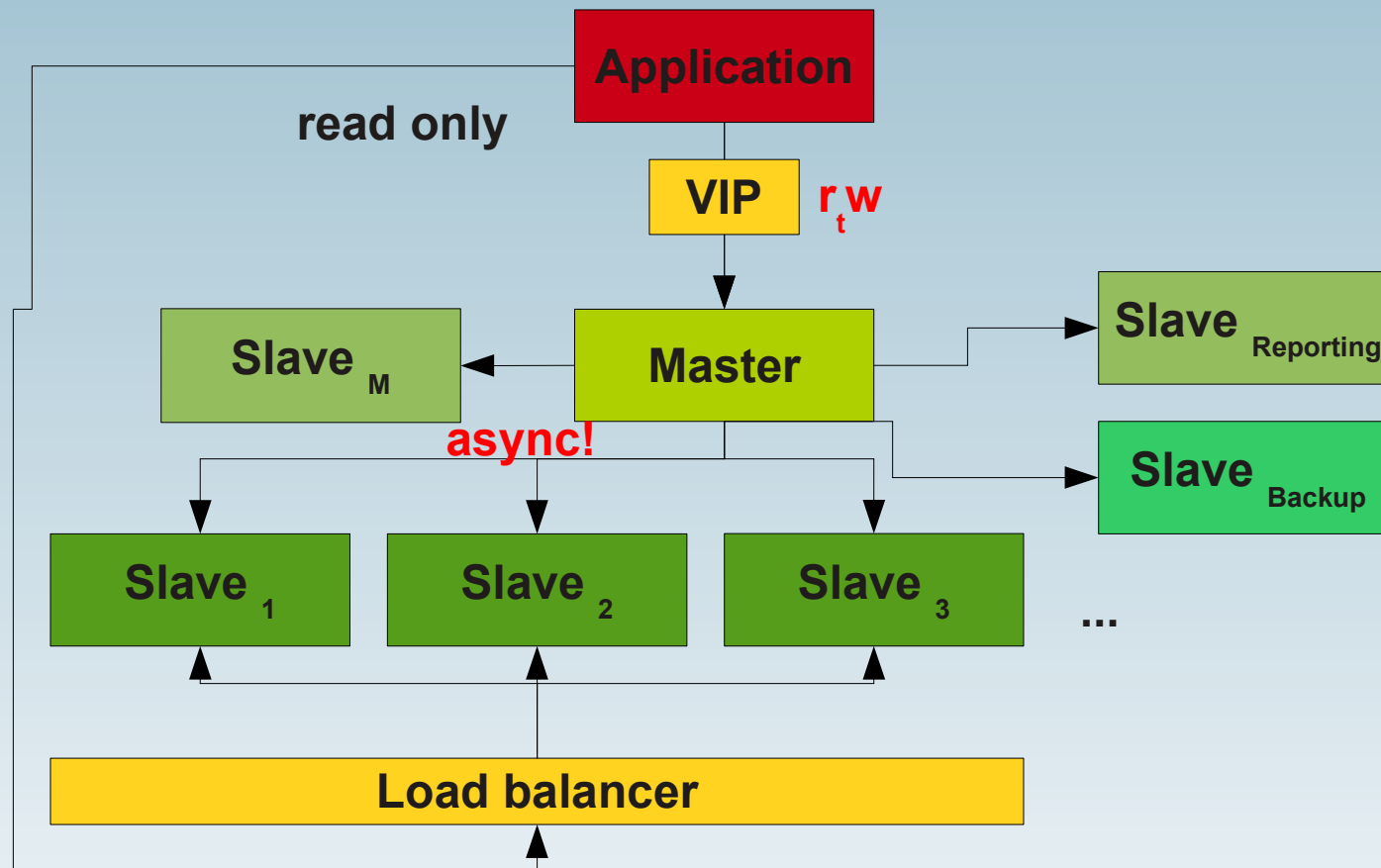
**Scale-Out**

# Master – Slave Replikation



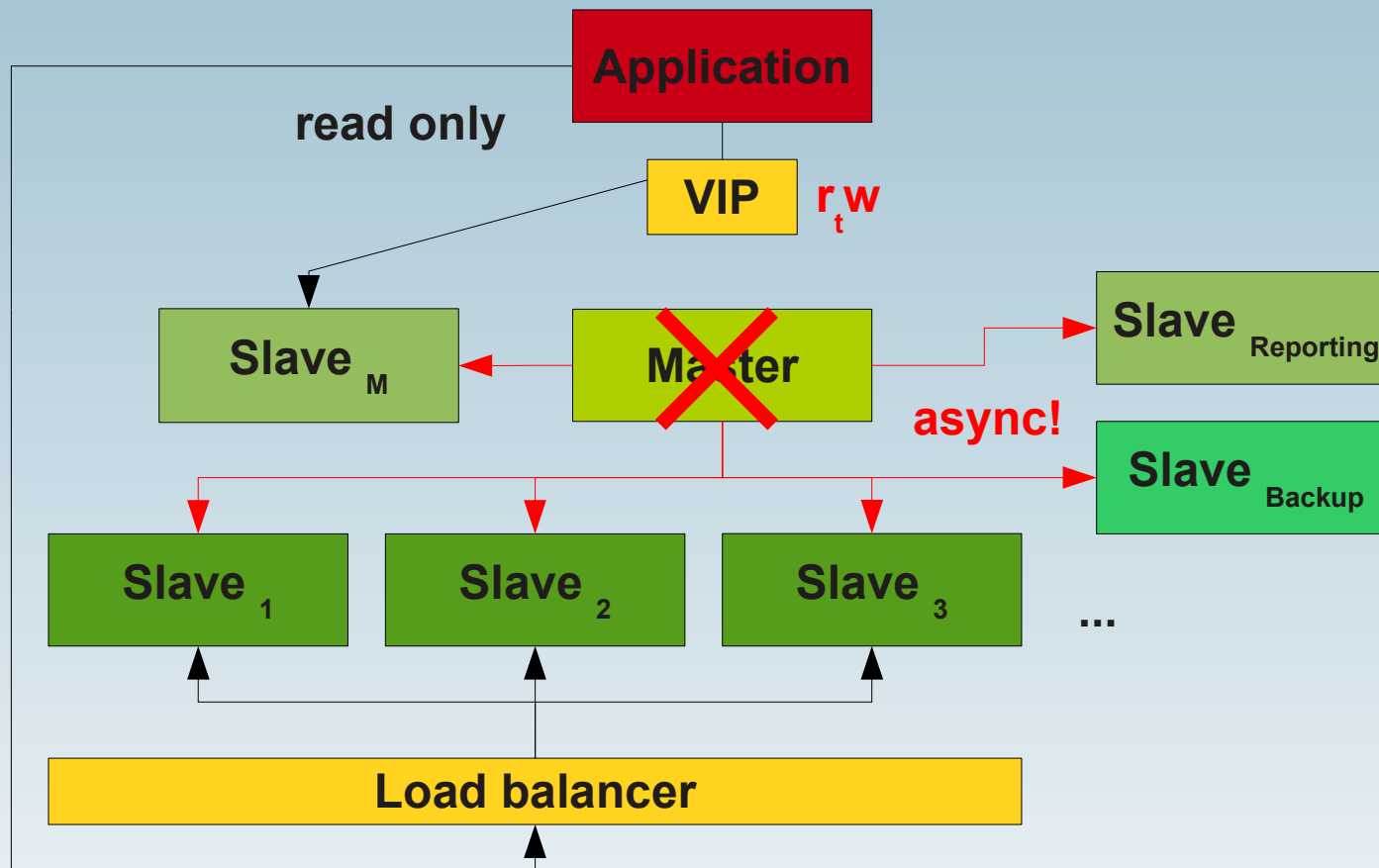
- **Wir brauchen:**
  - Binary Log
  - Server Id
  - User für die Replikation (auf dem Master)
  - Konsistentes Backup MIT Binary Log Position

# High-Availability mit Replikation www.fromdual.com



- Fail-over?

# Replikation Fail-over

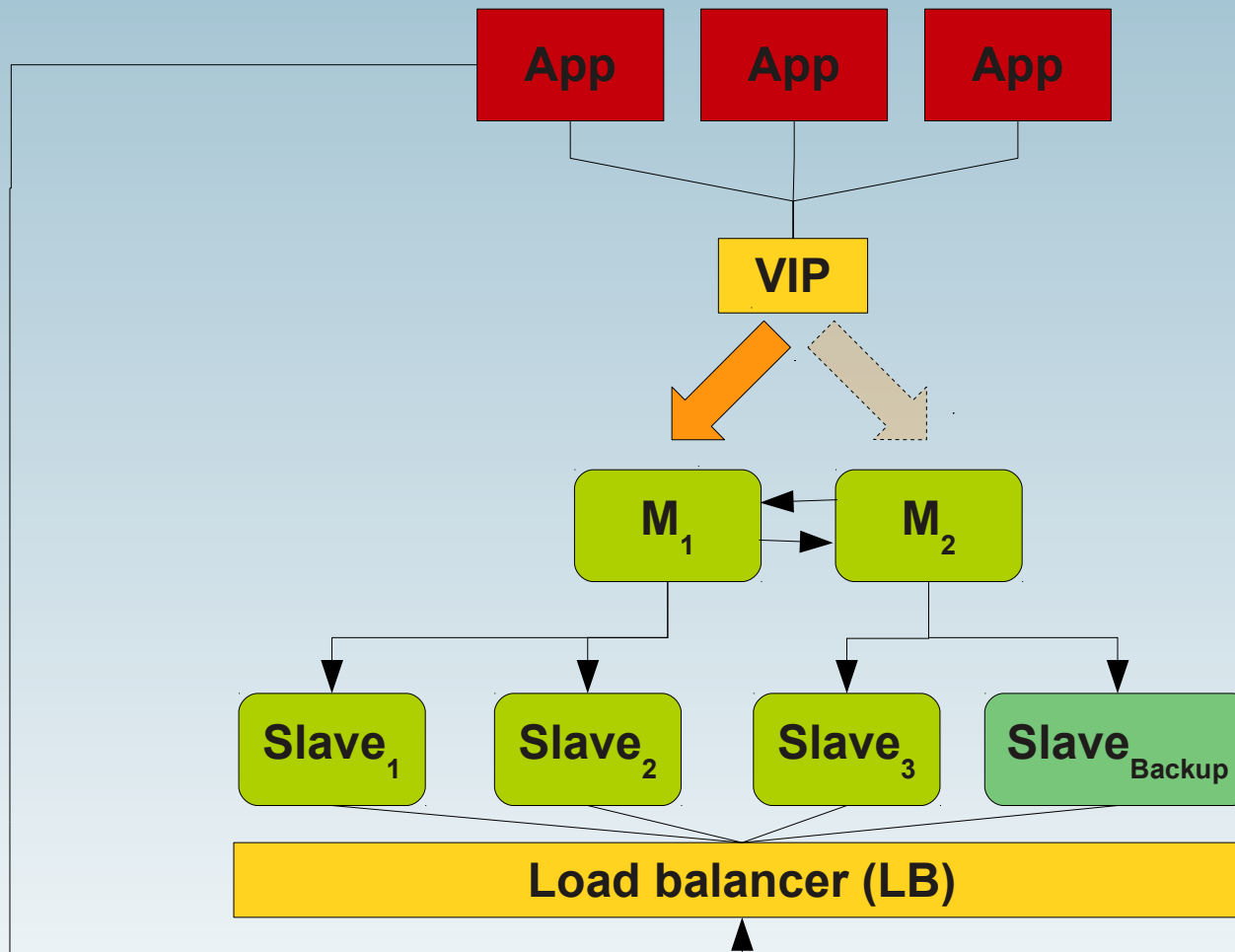


# Vorteile / Nachteile

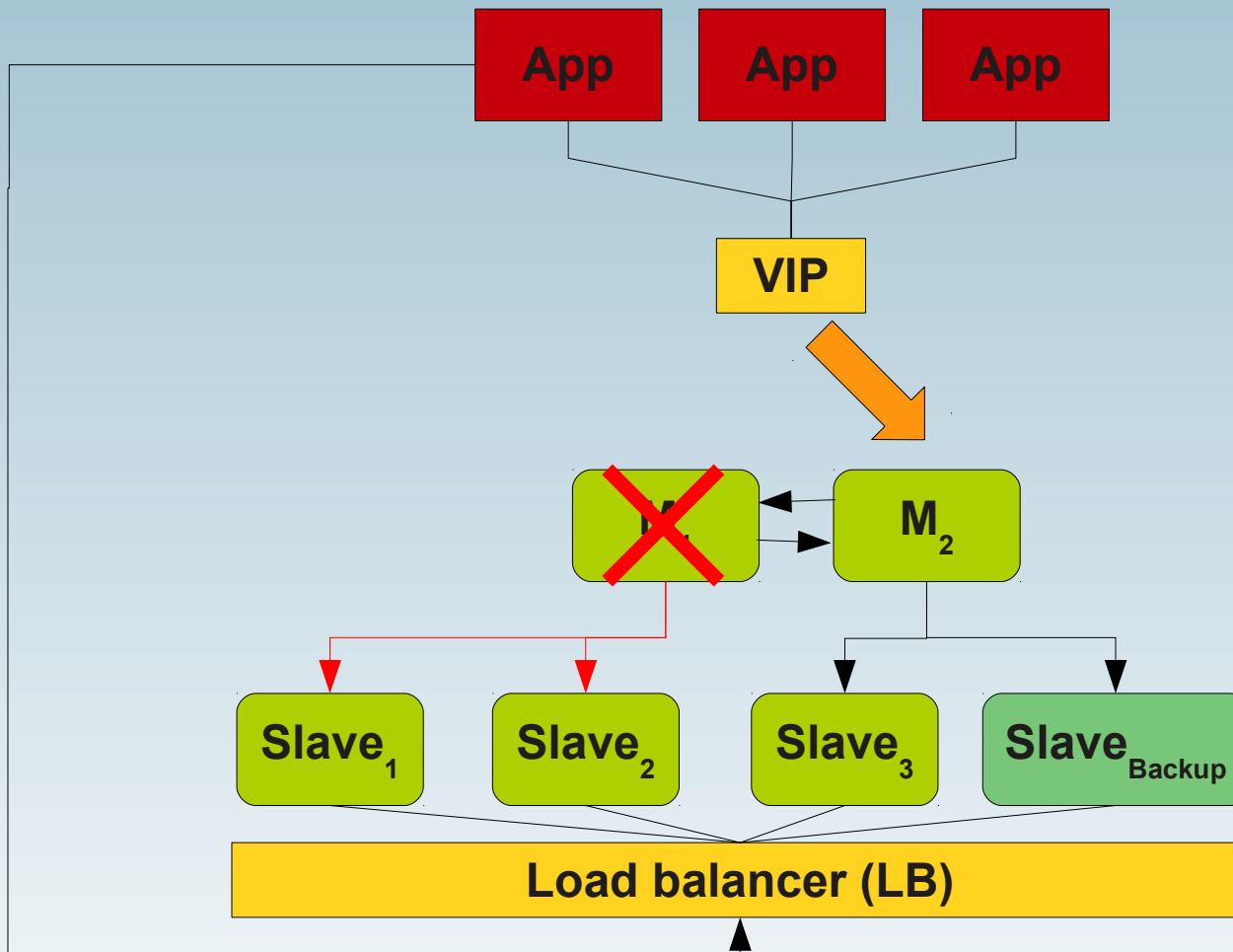
- Einfaches „standard“ Set-up
- Sehr gut wenn  $r \gg w$
- Fail-over Seite ist bereits warm/heiss!
- Zeitversatz Master/Slave (asynchron!)
- Slave kann hinterherhinken (Slave ist oft Flaschenhals)
- Daten **In**-konsistenz (`pt-table-checksum/pt-table-sync`)
- Wenn Master stirbt → welcher Slave wird neuere Master?
  - Switch → viel Arbeit, etwas heikel!
  - Es gibt Tools die helfen (MMM v1/v2, MHA, Tungsten, ...)



# Master-Master Replikation



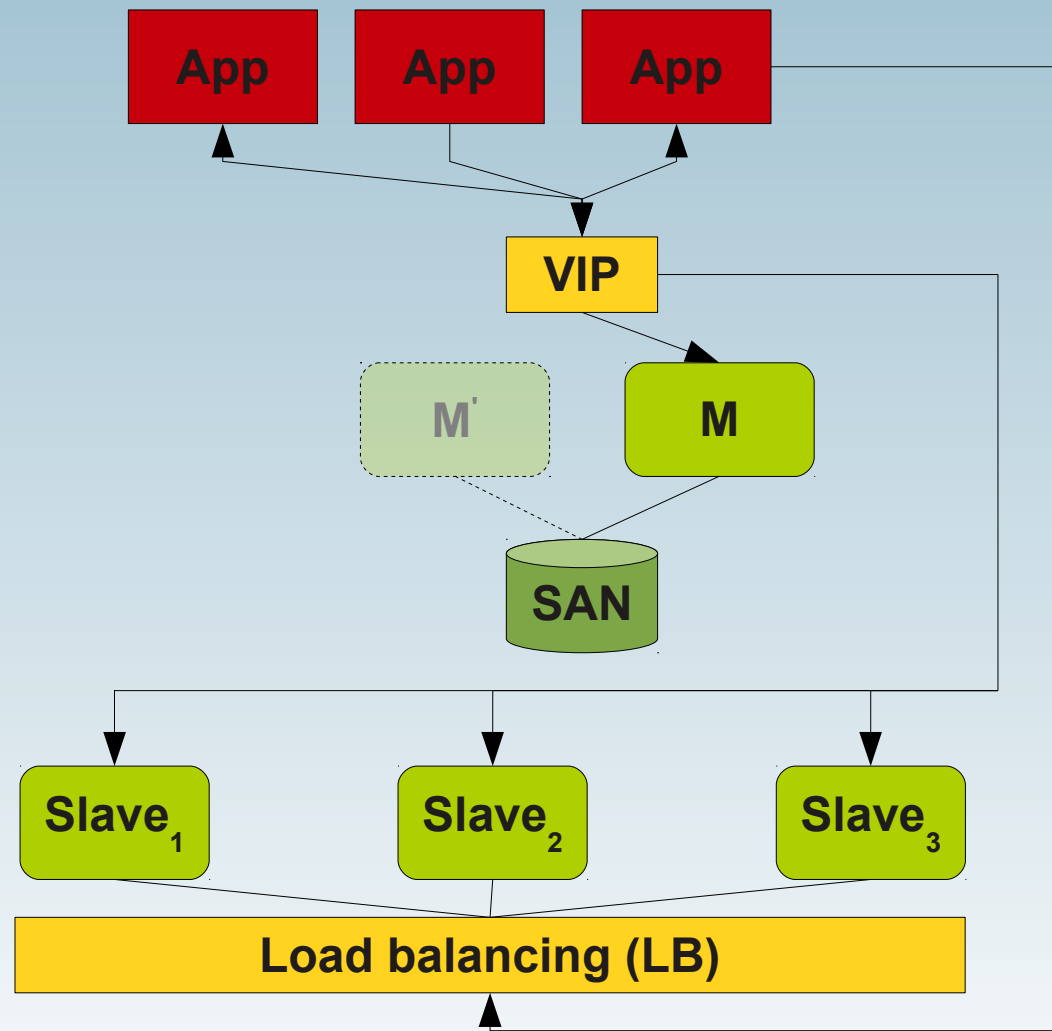
# Master-Master Replikation



# Vorteile / Nachteile

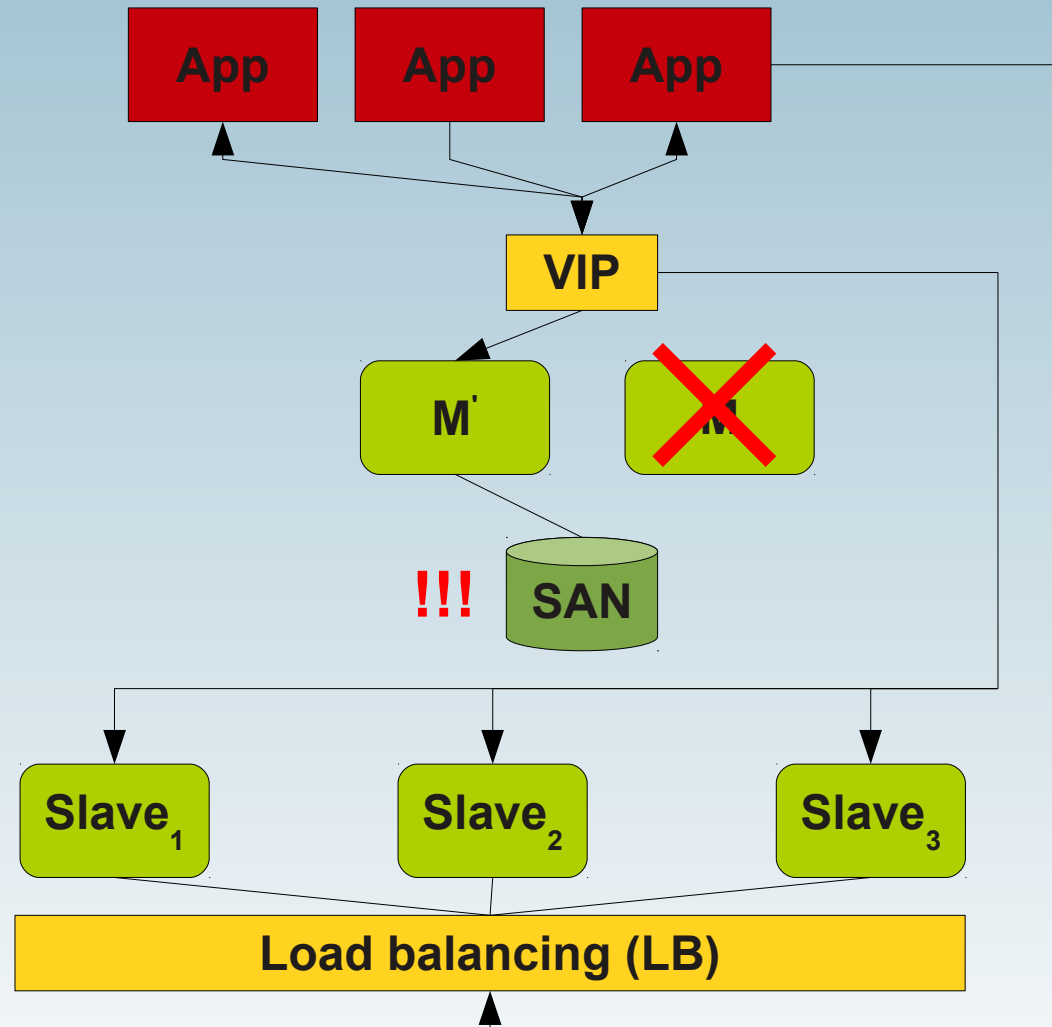
- Nur wenig komplexer als Master/Slave
- Sehr gut wenn  $r \gg w$
- Fail-over Seite ist bereits warm/heiss!
- Zeitversatz Master/Slave (asynchron!)
- Slave kann hinterherhinken (Slave ist oft Flaschenhals)
- Daten **In**-konsistenz (`pt-table-checksum/pt-table-sync`)
- Wenn Master stirbt, ist die Hälfte der Slaves „out of sync“!
- Vorsicht beim Schreiben auf beide Master!
- Man erhält dadurch NICHT mehr I/O Durchsatz!
- Ein wenig komplizierter (wieder-)aufzusetzen

# Aktiv/passiv fail-over mit SAN www.fromdual.com



# Aktiv/passiv fail-over mit SAN www.fromdual.com

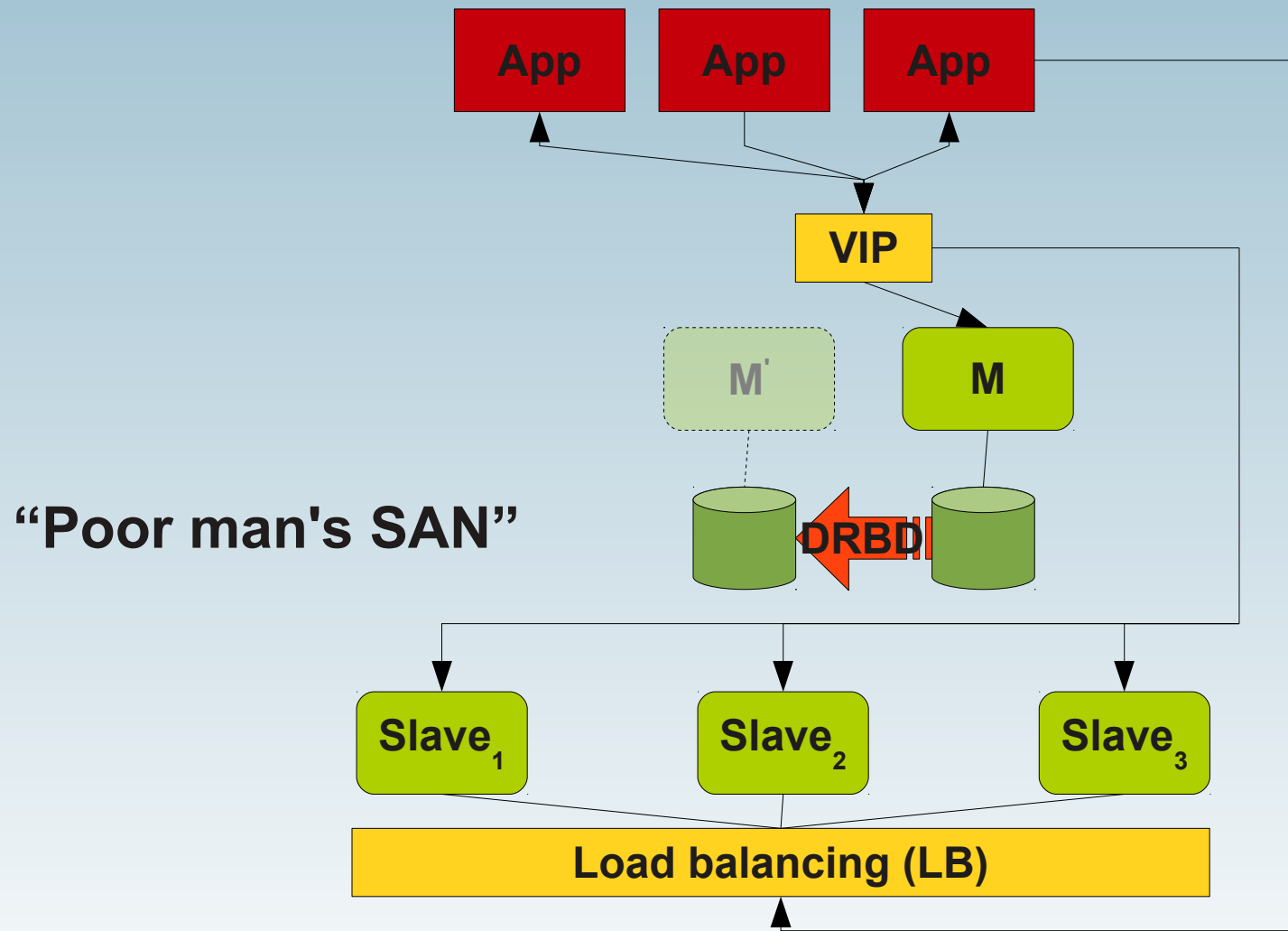
- SPOF 2!



# Vorteile / Nachteile

- Synchroner Replikation
- I/O Durchsatz hängt vom SAN (I/O System) ab
- Keine Daten **IN**-Konsistenzen möglich
- Nur eine mögliche Datenquelle
- Slaves werden automatisch und sauber geschwenkt
- SAN und Filesystem sind SpoFs!
- Teuer wenn SAN noch nicht vorhanden ist.
- SAN's sind nicht einfach richtig zu betreiben!
- Andere Seite ist kalt nach Fail-over!
- Hälfte der Hardware idelt
- Wesentlich komplexer aufzusetzen

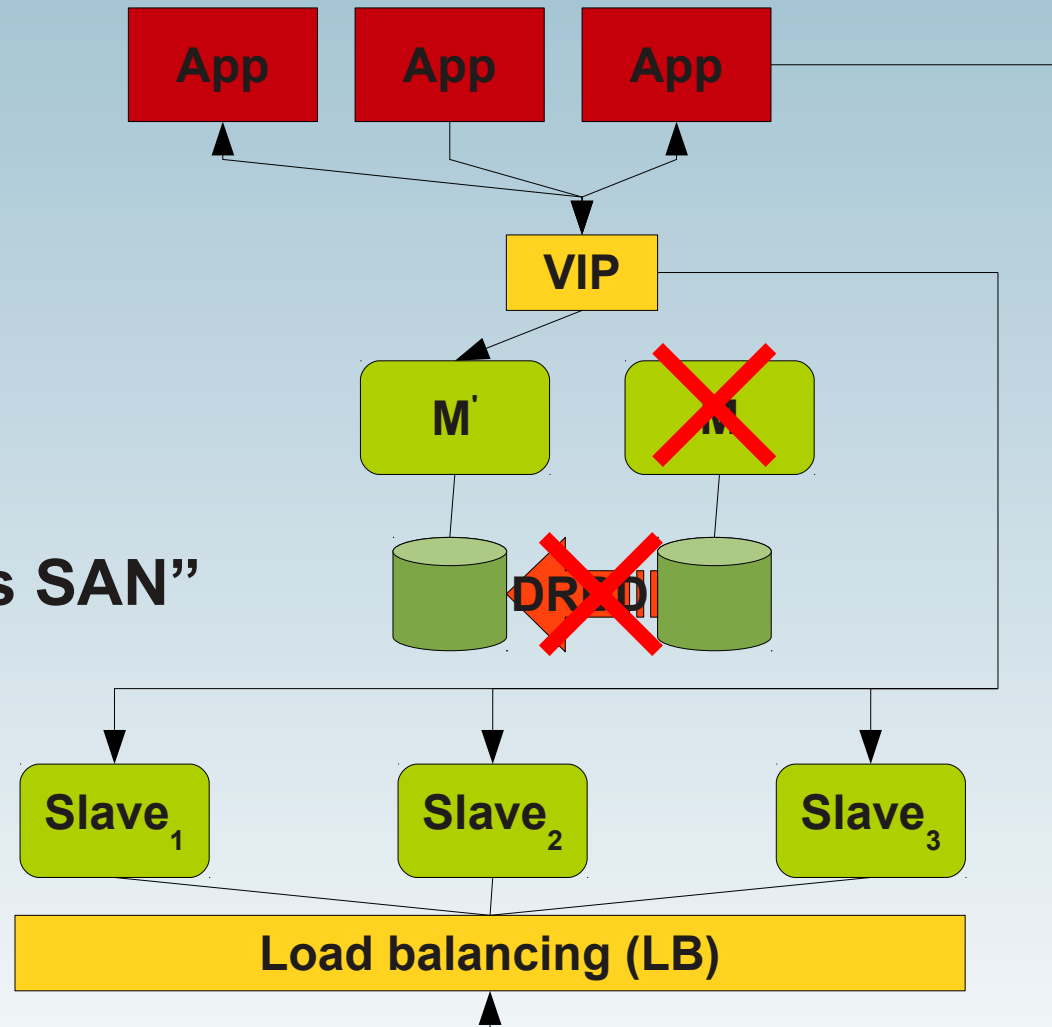
# Aktiv/passiv fail-over mit DRBD



# Activ/passiv fail-over mit DRBD

- SPOF 1!

“Poor man's SAN”

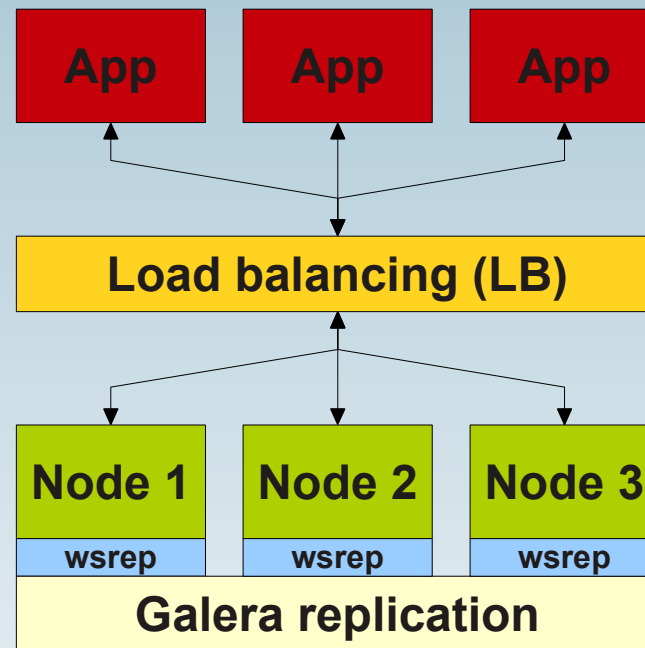




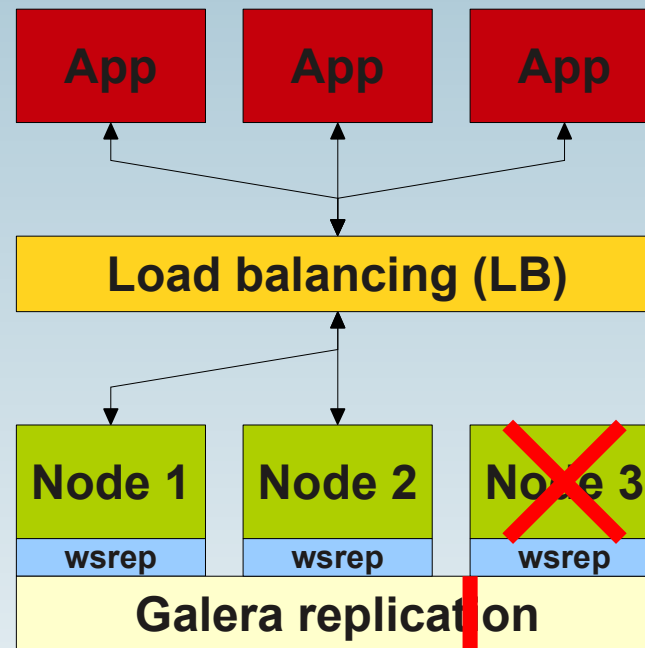
# Vorteile / Nachteile

- **Synchrone Replikation**
- **Keine Daten **IN**-Konsistenzen möglich**
- **Nur eine mögliche Datenquelle**
- **Slaves werden automatisch und sauber geschwenkt**
- **Filesystem ist SpoF!**
- **I/O Durchsatz tendenziell geringer als mit SAN**
- **Andere Seite ist kalt nach Fail-over!**
- **Hälfte der Hardware idelt**
- **Wesentlich komplexer aufzusetzen**

# Galera Cluster für MySQL



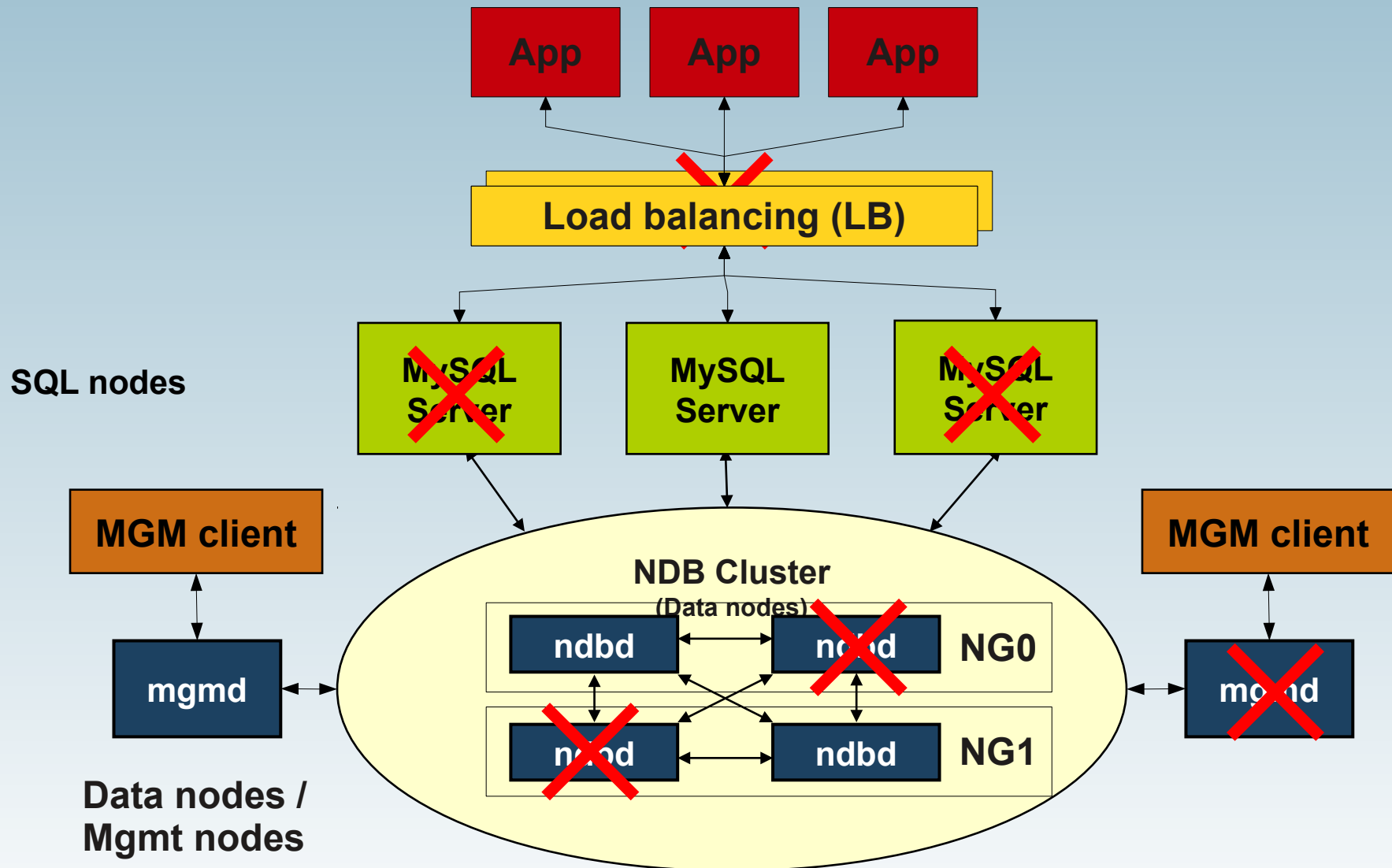
# Galera Cluster für MySQL



# Vorteile / Nachteile

- **Synchrone Replikation**
- **Aktiv-aktiv multi-Master Topology**
- **Lesen und Schreiben auf alle Cluster-Knoten**
- **Automatische Knotenverwaltung**
- **Echtes paralleles Replizieren auf Zeilenebene**
- **Kein Hinterherhinken des Slaves**
- **Keine verlorene Transaktionen**
- **Lese-Skalierbarkeit (read Scale-Out!) und mehr Schreiben (+ SSD)**
- **Wartung im „laufenden Betrieb“ möglich**
- **Basiert auf InnoDB Storage Engine (nur auf InnoDB!)**
- **Galera Binaries (nicht Oracle/MySQL)**
- **Achtung vor Hots-Spots auf einzelnen Zeilen (z. B. Sequenzen-Tabelle)**
- **Dadurch höhere Wahrscheinlichkeit von Deadlocks**
- **Voll-Synchronisation (SST) blockiert Lesen und Schreiben → 3 nodes**

# MySQL (NDB) Cluster



# Vorteile / Nachteile

- **Synchrone Replikation**
- **Nur eine mögliche Datenquelle**
- **Keine Daten **IN**-Konsistenzen möglich**
- **Extrem hoher Durchsatz (wenn richtig gemacht)**
- **Skaliert sehr gut für Lesen UND Schreiben (wenn richtig gemacht)**
- **Wartung im „laufenden Betrieb“ möglich**
- **Kein drop-in Ersatz für InnoDB/MyISAM!**
- **Hoher Bedarf an RAM und Netzwerk**
- **Mindestens 3 Server (besser 4) sind erforderlich.**
- **Neue Datenbank zu lernen (MySQL Cluster != MySQL!)**
- **Komplexer aufzusetzen und zu betreiben als normales MySQL**
- **Schlecht für Joins (Push Down Joins in v7.2)**

# Q & A



[www.fromdual.com](http://www.fromdual.com)



Fragen ?

Diskussion?

**Wir haben Zeit für ein High-Availability Audit...!**

- **FromDual bietet neutral und unabhängig:**
  - **Beratung**
  - **Remote-DBA**
  - **Support für MySQL und Galera Cluster**
  - **Schulung**

[www.fromdual.com/presentations](http://www.fromdual.com/presentations)