



Weltweite Produktionsdatenverwaltung mit MySQL-Replikation

DOAG Konferenz 2015, Nürnberg

Oli Sennhauser

Senior MySQL Consultant, FromDual GmbH

oli.sennhauser@fromdual.com



www.fromdual.com

Über FromDual GmbH

Support



Beratung



remote-DBA



Schulung

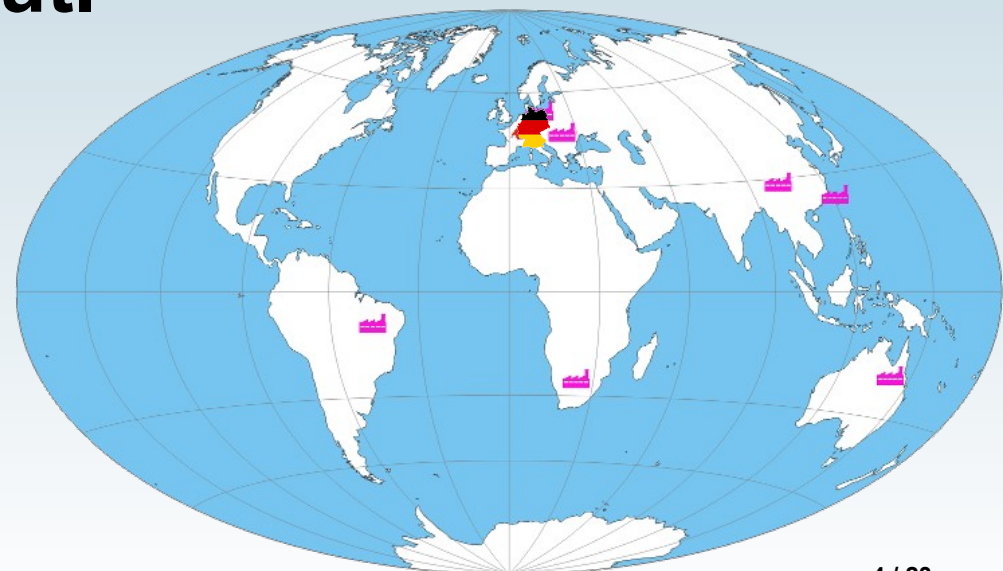


Weltweite Produktionsdatenverwaltung mit MySQL

- **Ausgangslage**
- **Probleme die sich stellen**
- **Wer darf welche Daten kriegen?**
- **Wie werden Daten verteilt?**
- **Produktionsdaten zurück?**
- **MySQL multi-Source Replikation**
- **Bertriebsverantwortung**
- **Hochverfügbarkeit**
- **Sensitive Daten**
- **Wer darf welche Daten sehen?**
- **MySQL Row Filterung**
- **Nachträgliche Forderungen**

Ausgangslage

- Immer mehr deutsche Unternehmen lagern Ihre Produktion ins nahe und ferne Ausland aus.
- Das hat organisatorische und sicherheitstechnische Überlegungen zur Folge.
- Insbesondere dann, wenn man dem Partner im Ausland nicht 100% traut.
- Und staatliche und private Organisationen Begehrlichkeiten zeigen...

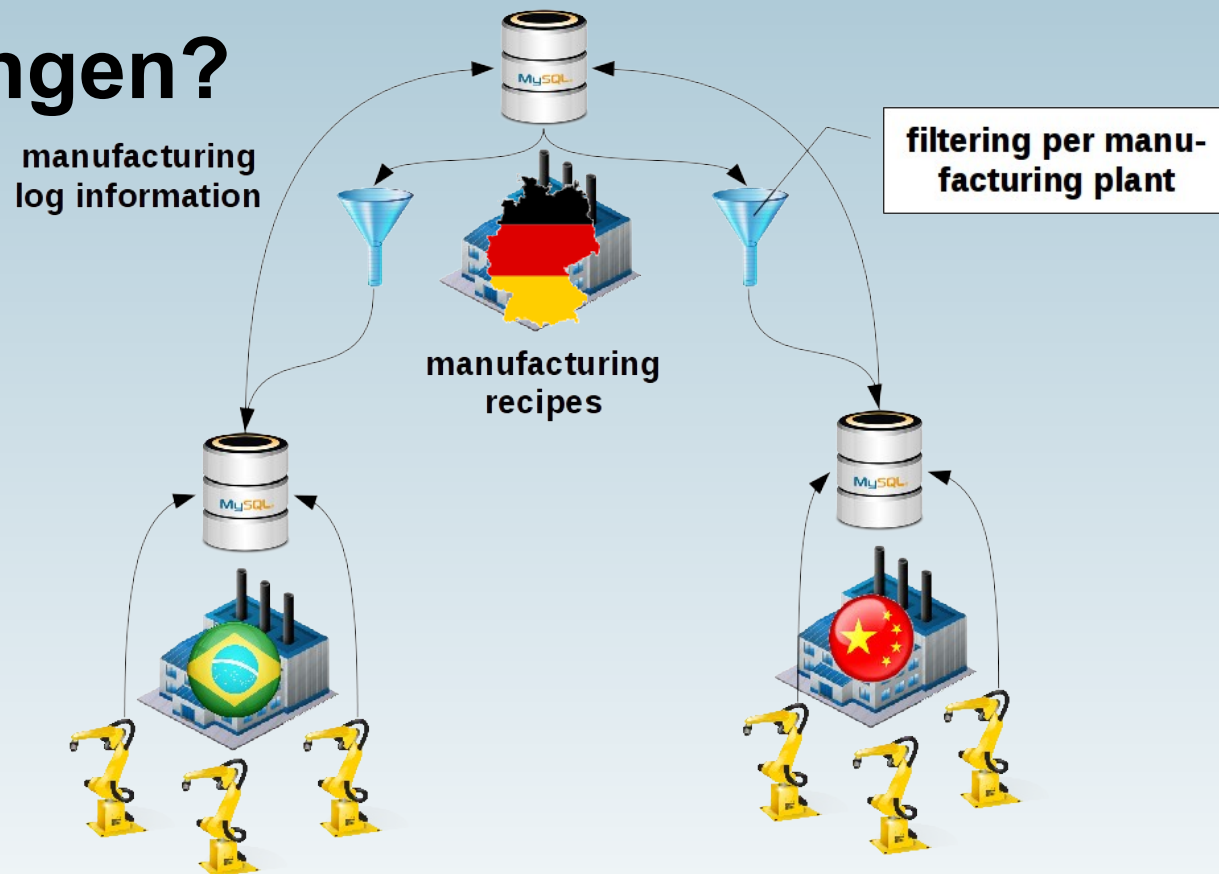


Probleme, die sich stellen

- **Wer darf welche Daten kriegen?**
- **Wie werden die Daten verteilt?**
- **Wie kriegen wir die Produktionsdaten wieder zurück in die Zentrale?**
- **Wer hat die Verantwortung über welche Komponenten?**
- **Wie stellen wir sicher, dass geheime Daten nicht nachträglich eingefordert werden können?**

Wer darf welche Daten kriegen?

- Welche Daten?
- Anforderungen?

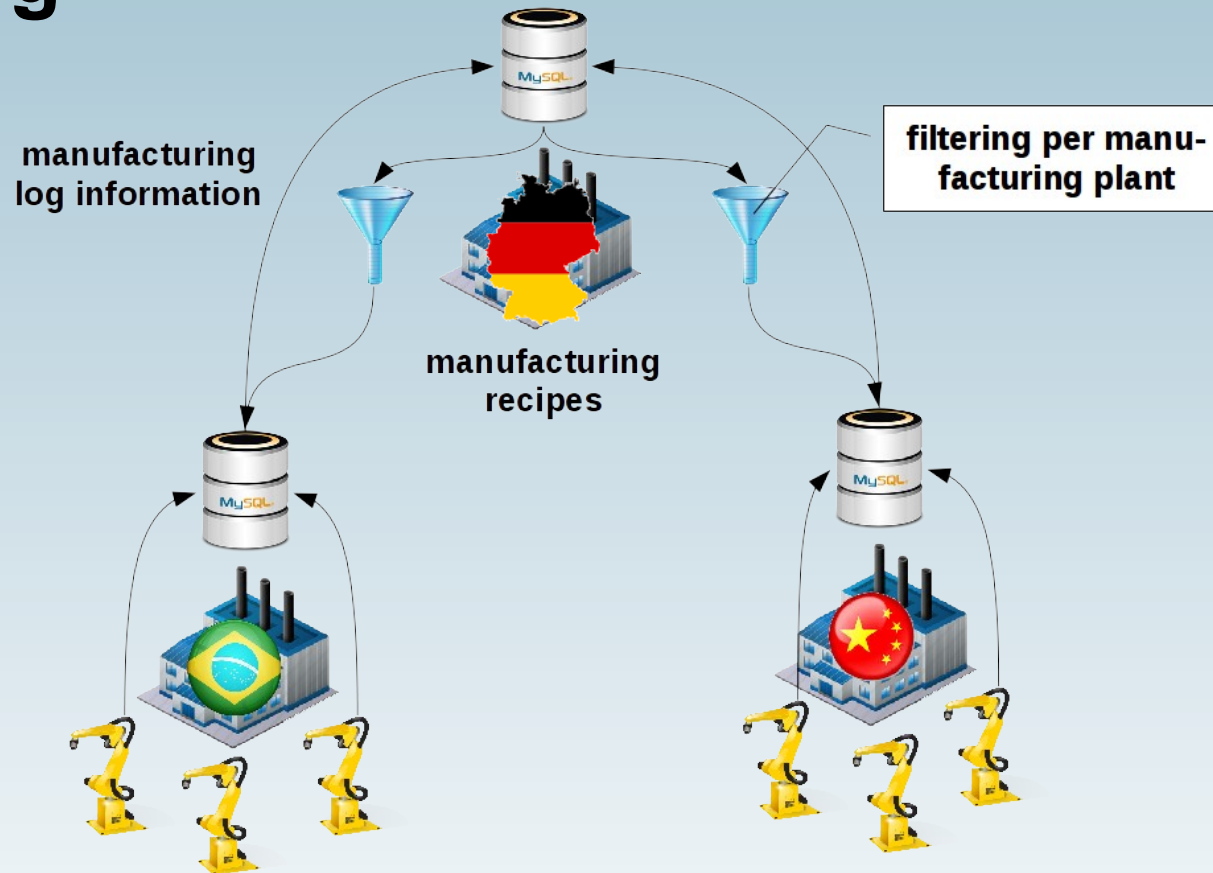


Wer darf welche Daten kriegen?

- **Welche Daten haben wir überhaupt?**
 - **Produktions-Rezepturen**
 - **Daten, die verbaut werden**
 - **Produktions-Messdaten**
- **Anforderungen**
 - **Land A soll NICHT Daten von Land B sehen!**
 - **Zentrale soll alle Messdaten erhalten**
 - **Verbaute Daten nur in kleinen Häppchen verteilen (Missbrauch).**

Wie werden Daten verteilt?

- **Hinweg**

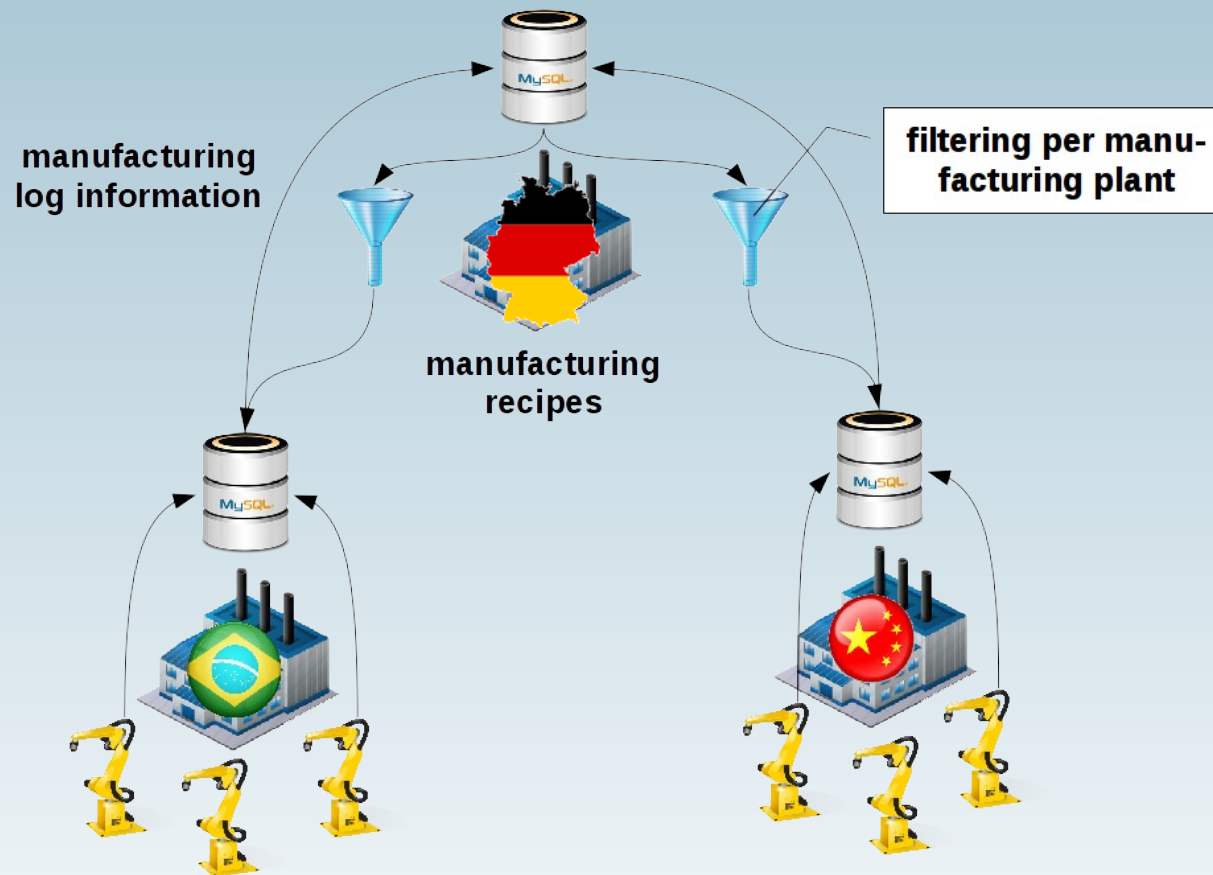


Wie werden Daten verteilt?

- **Hinweg, 2 Möglichkeiten:**
 - **Applikatorisch :-)**
 - **MySQL Master/Slave Replikation**

- **Kunde hat MySQL M/S Replikation gewählt**
 - **Einfach und robust**
 - **Schnell zu implementieren**
 - **Standard out-of-the-box Verfahren**

Produktionsdaten zurück?



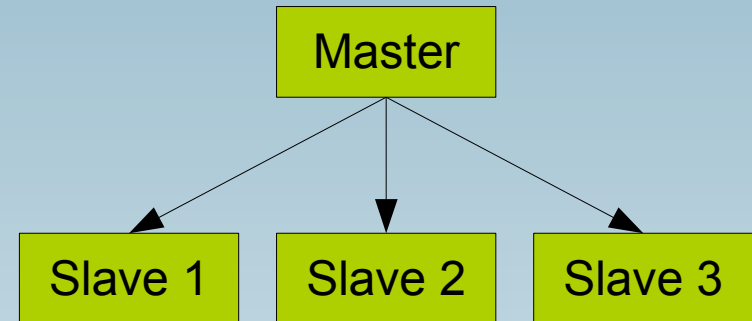
Produktionsdaten zurück?

- **Wie kriegen wir die Produktionsdaten wieder zurück in die Zentrale?**
- **Rückweg**
 - Applikatorisch :-)
 - MySQL multi-Source Replikation (5.7)
- **Kunde hat vorerst applikatorisch gewählt**
 - da recht einfache Logik

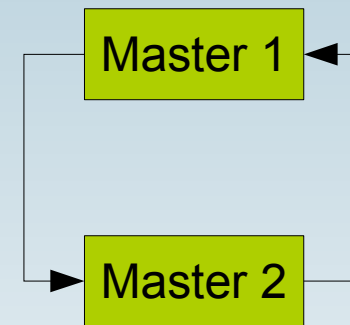
```
REPLACE INTO zentrale
SELECT * FROM land WHERE ts > <letzter_lauf>;
```
 - MySQL multi-Source Replikation noch nicht GA war
 - Sommer 2015
 - Betrieb nicht ganz trivial ist

multi-Source Replikation

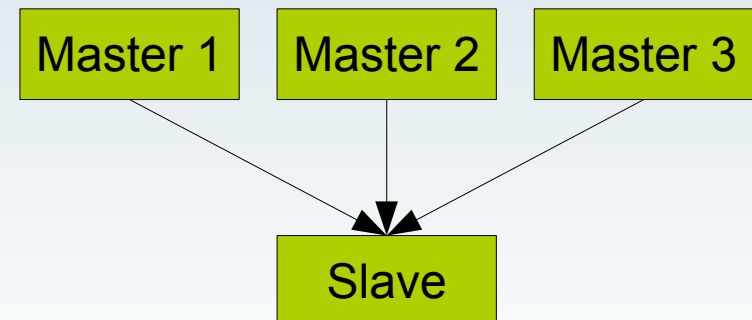
- Was wir bisher konnten:
Master/Slave



- und Master/Master:

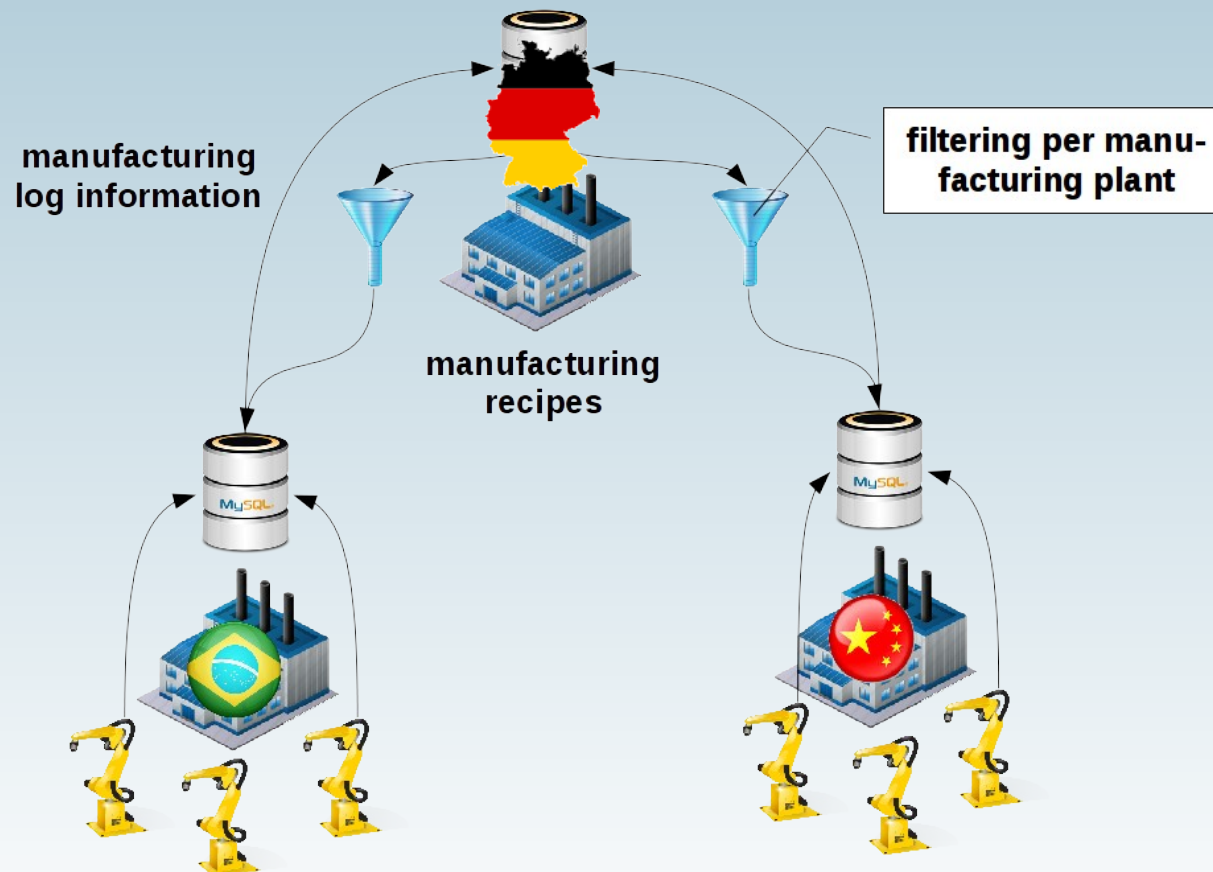


- neu ist multi-Source:



Betriebsverantwortung

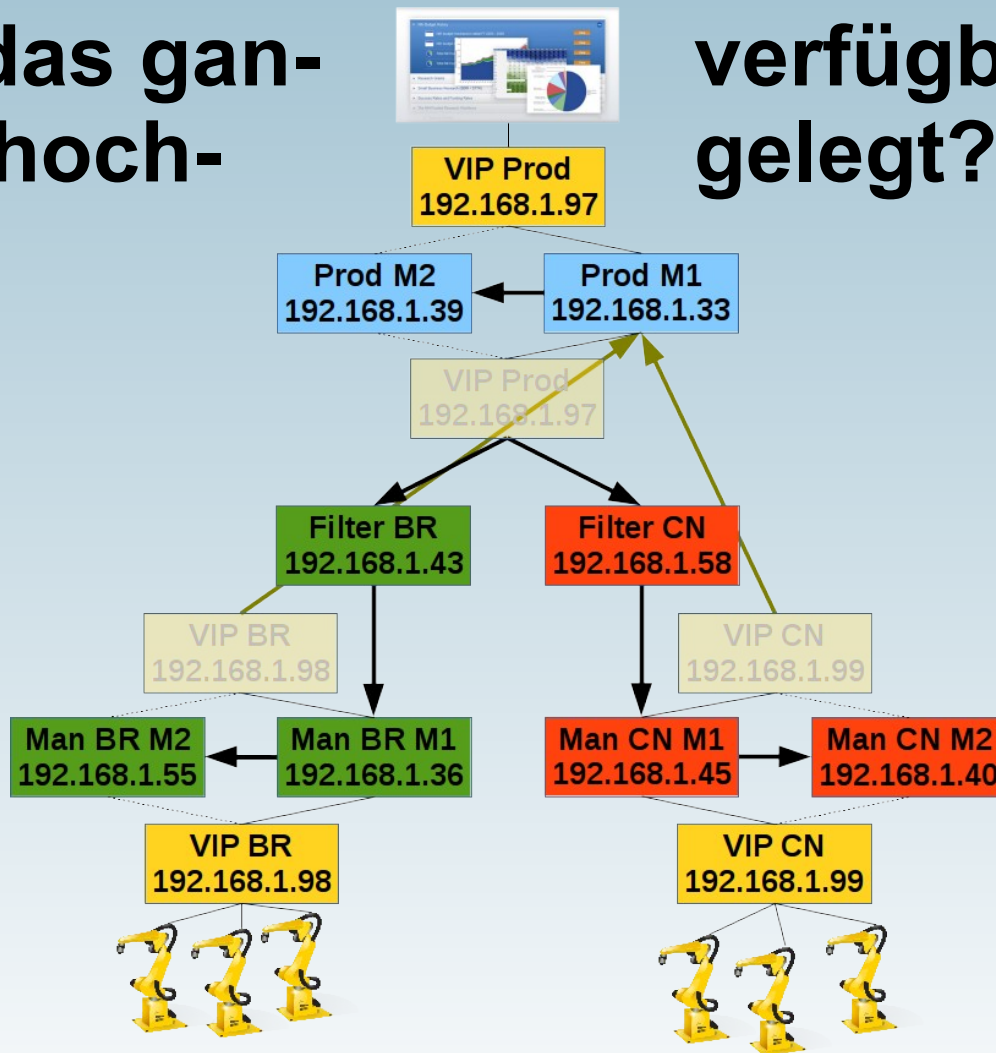
- Wer hat die Verantwortung über welche Komponenten?



Hochverfügbarkeit

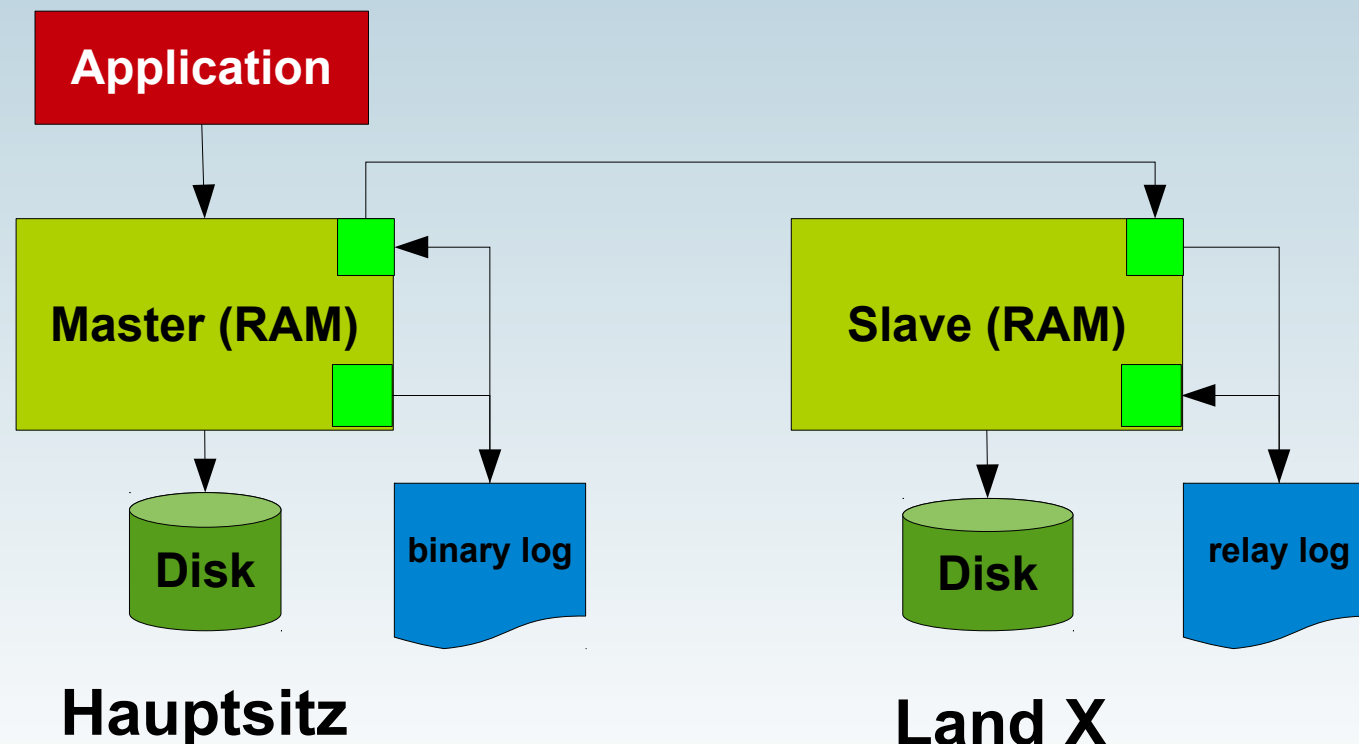
Wie wir das ganze noch hoch-

verfügbar ausgelegt?



Sensitive Daten

- Wie stellen wir sicher, dass sensitive Daten, das Haus nicht verlassen?
- Wie funktioniert MySQL Replikation?

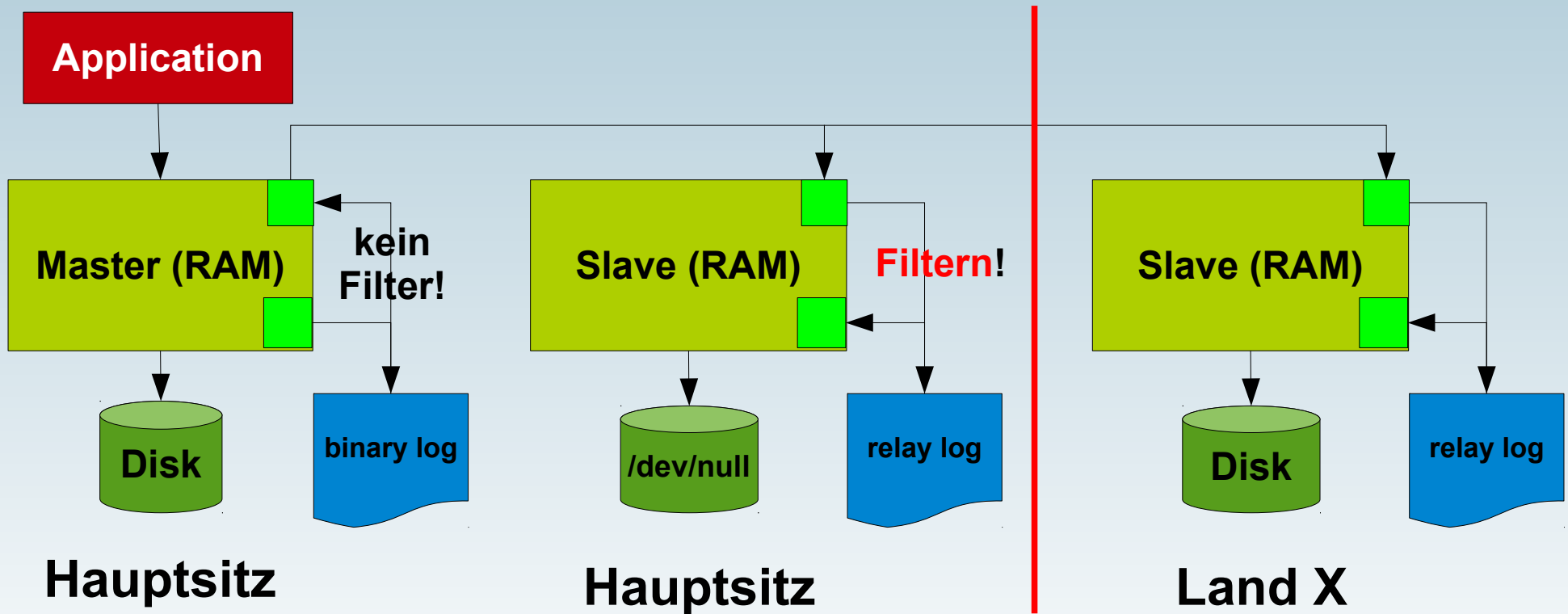


Wer darf welche Daten sehen?

- **Land A soll nicht Daten von Land B sehen!**
 - d.h. Land A soll nicht Daten von Land B kriegen!!!
 - Sonst kann der gewiefte Admin die auch sehen...!
- **Wie? MySQL Replikationsfilterung!**
 - Master: `binlog-db, binlog-ignore-db`
 - Slave: `replicate-do-db, replicate-ignore-db, replicate-do-table, replicate-ignore-table`
- **Problem: Filtern auf dem Master :-)**
 - zwar nicht in Replikation aber
 - auch nicht verfügbar für Point-in-Time-Recovery!

Lösung?

- **Blackhole Storage Engine!**
 - `/dev/null`



MySQL Row Filterung?

- **Schema Filterung?** ✓
- **Tabellen Filterung?** ✓
- **d.h. aber, wir müssen unsere Applikation ev. umbauen?!?**
 - **1 Schema pro Land**
 - **Vielleicht gar nicht so schlechte Idee, aber Arbeit!**
- **Row Filterung?**
 - **MySQL :-)**
 - **aber ...**

MySQL Row Filterung?

- MySQL kann das nicht.
- Aber, Lego-Kiste auf...

```
CREATE TRIGGER filter_row
BEFORE INSERT ON manufacturing_data
FOR EACH ROW
BEGIN

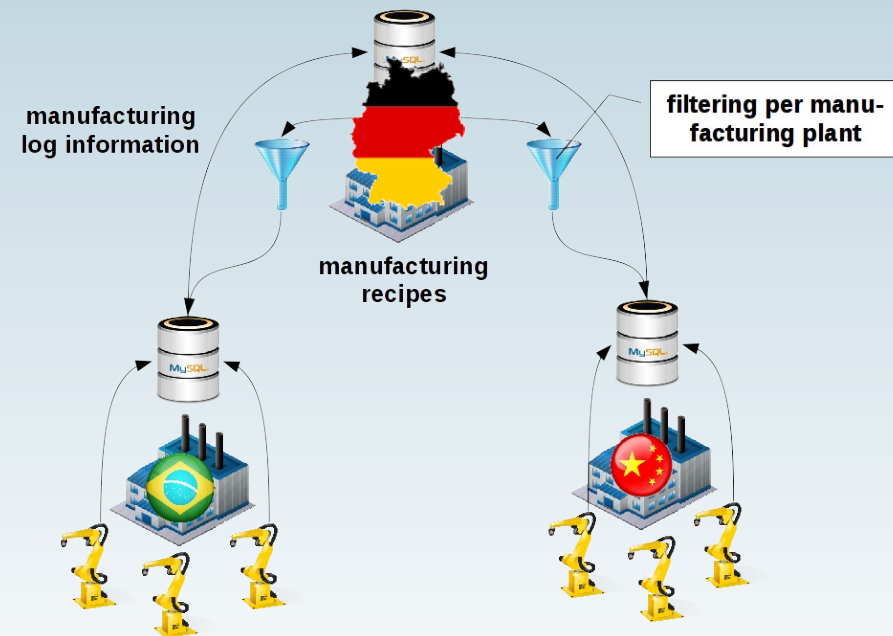
    IF ( NEW.manufacture_plant != 'China' ) THEN

        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT      = 'Row was filtered out.'
          , CLASS_ORIGIN      = 'FromDual filter trigger'
          , SUBCLASS_ORIGIN   = 'filter_row'
          , CONSTRAINT_SCHEMA = 'erp'
          , CONSTRAINT_NAME   = 'filer_row'
          , SCHEMA_NAME       = 'erp'
          , TABLE_NAME       = 'manufacturing_data'
          , COLUMN_NAME       = ''
          , MYSQL_ERRNO       = 1644
        ;
    END IF;
END;
```



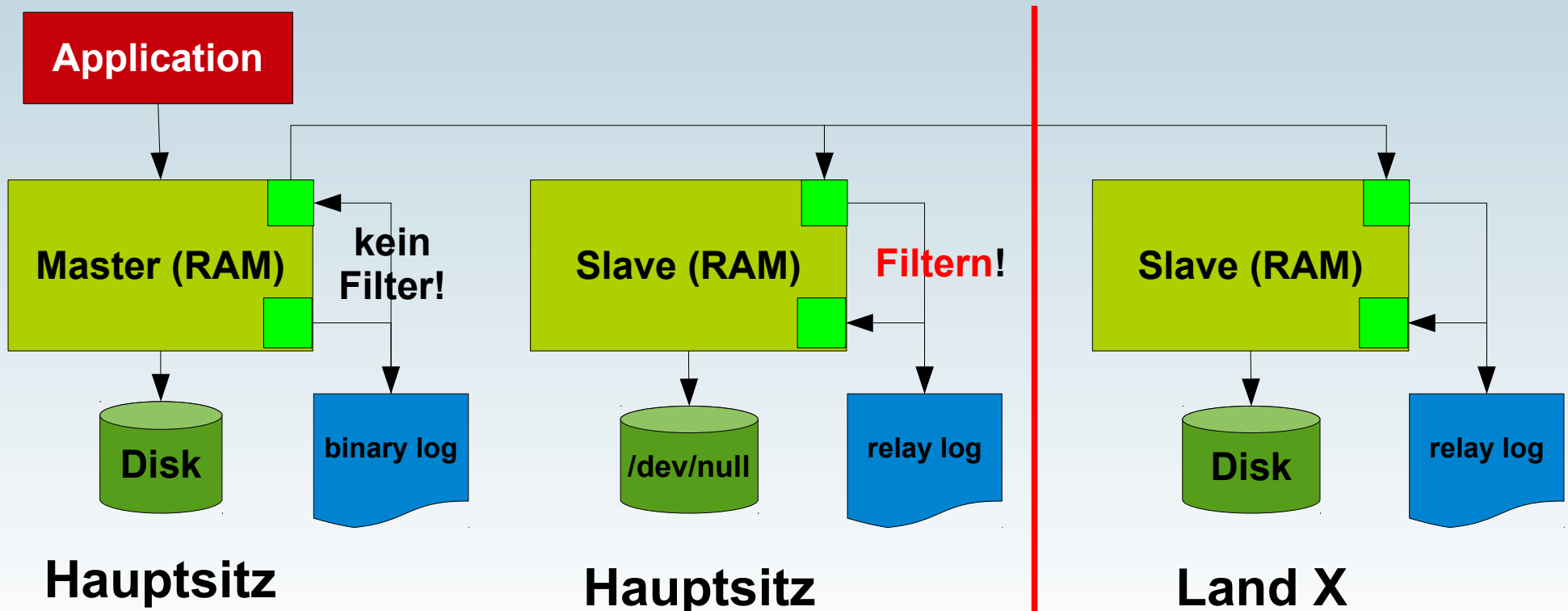
Nachträgliche Forderungen

- **Wie stellen wir sicher, dass geheime Daten nicht nachträglich eingefordert werden können?**
 - **„Es“ will da mitgucken/mithören!**



Nachträgliche Forderungen

- Gar nicht ausliefern
- Nur so wenig wie möglich
- Asap wieder sauber löschen



<http://fromdual.com/controlling-worldwide-manufacturing-plants-with-mysql>

Schulung: MySQL Operations und Hochverfügbarkeitsarchitekturen

ca. 6 x im Jahr (Berlin und Essen)

Stand 308

Q & A



Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
 - **Beratung**
 - **Remote-DBA**
 - **Support für MySQL, Galera, Percona Server und MariaDB**
 - **Schulung**

www.fromdual.com/presentations