



Der Datenbank-Backup ist gemacht - was nun ?

SLAC 2015, Berlin

Jörg Brühe

Senior Support Engineer, FromDual GmbH

joerg.bruehe@fromdual.com



FromDual GmbH

www.fromdual.com



Support



Beratung



remote-DBA



Schulung



Wir suchen noch:



**MySQL Datenbank Enthusiast/in für
Support / remote-DBA / Beratung**

Zur Person

- **Entwicklung verteiltes SQL-DBMS:**
Unix-Portierung,
Anschluss Archivierungs-Tools (ADSM, NetWorker)
- **MySQL Build Team:**
Release-Builds inkl. Tests, Paketierung, Skripte, ...
- **DBA:**
MySQL für eine Web-Plattform
(Master-Master-Replikation)
- **Support-Ingenieur (FromDual):**
Support + Remote-DBA für MySQL / MariaDB / Percona
mit oder ohne Galera Cluster

Inhalt

Datenbank-Backup – welcher Zweck?

Tauglichkeit des Backup, Verifikation

Echtdaten vollständig nutzen

Dem Datenschutz genügen

Material für die Entwicklung

Automatisierung

➔ **Datenbank-Backup – welcher Zweck?**

Tauglichkeit des Backup, Verifikation

Echtdaten vollständig nutzen

Dem Datenschutz genügen

Material für die Entwicklung

Automatisierung

Zweck des DB-Backup

Gilt für beliebige Datenbanksysteme:

- **Firma / Mensch ist auf Daten angewiesen (finanziell, juristisch oder emotional)**
- **Datenverlust kann Betrieb verhindern**
- **„Wenn die Daten nicht gesichert wurden, dann sind sie auch nicht wichtig“**

Schadensursachen z.B. ...

- **Hardware-Defekte**
Laufwerk, Controller, Netzteil, ...
- **Software-Defekte**
DBMS, Betriebssystem, Anwendung, ...
- **Fehlbedienung / Sabotage**
`delete from T; commit; / rm -fr /`
- **Katastrophe**
Brand, Hochwasser, Unfall, ...

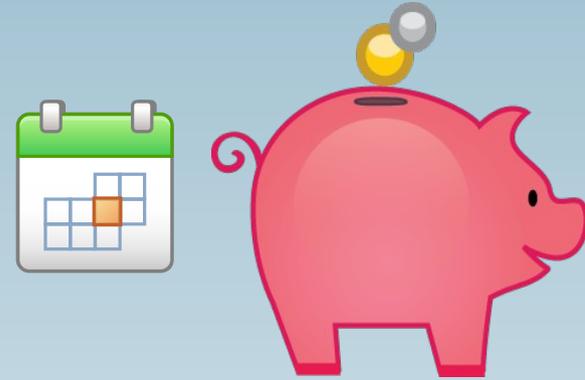
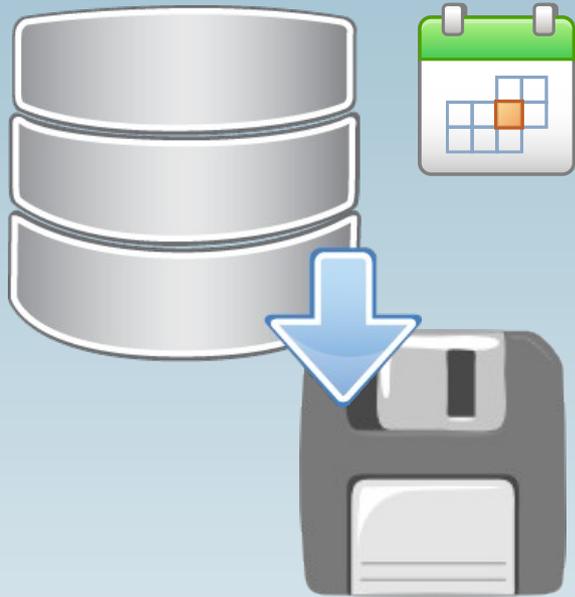
Risiko-Annahmen

Berücksichtigte Ereignisse

- **Individuelle Entscheidung**
- **Abhängig vom Geschäftszweck**
- **Abhängig von Technik und Lage**

**Setzen Anforderungen an Backup
z.B. Offsite-Speicherung nötig ja/nein?**

Backup wie Versicherung ...



Backup wie Versicherung (1)

Regelmäßiger

**Arbeitsaufwand
(Backup)**

**Geldaufwand
(Beitragszahlung)**

soll im Schadensfall

**Wiederherstellung
der Daten**

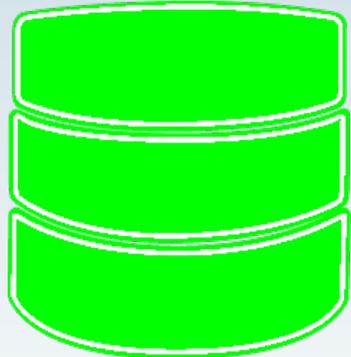
**Wiederbeschaffung
der Sache**

ermöglichen

(Daten noch da)

(genug Geld da)

Backup wie Versicherung ...



Backup wie Versicherung (2)

**Recovery
versagt**

**Versicherung
zahlt nicht**

wenn Schadensursache

**in der
Backup-Politik**

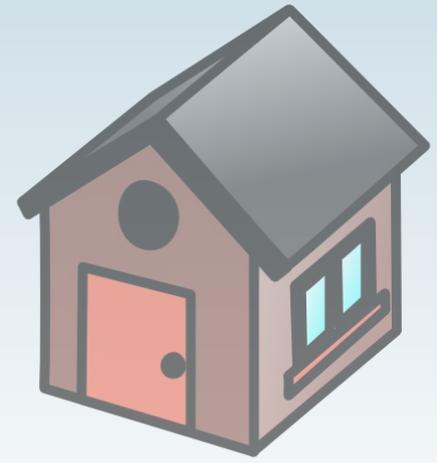
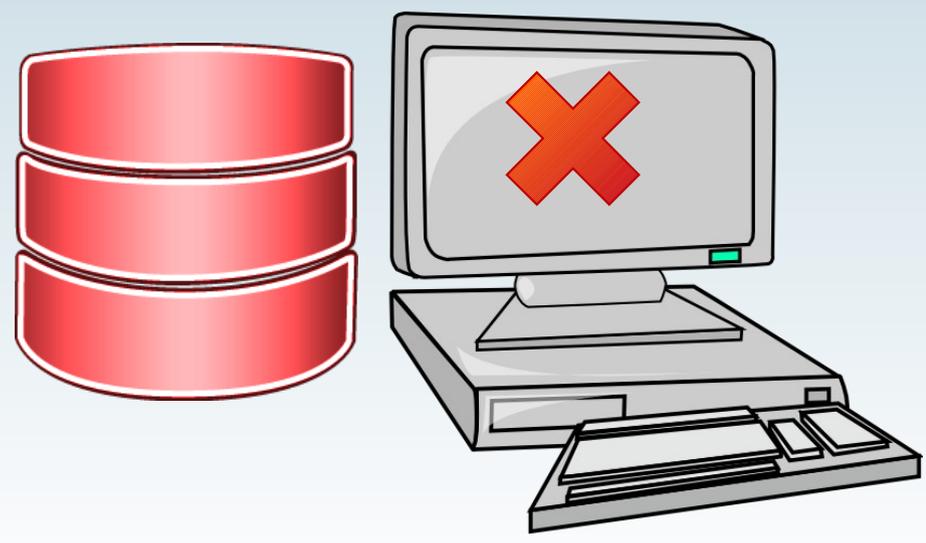
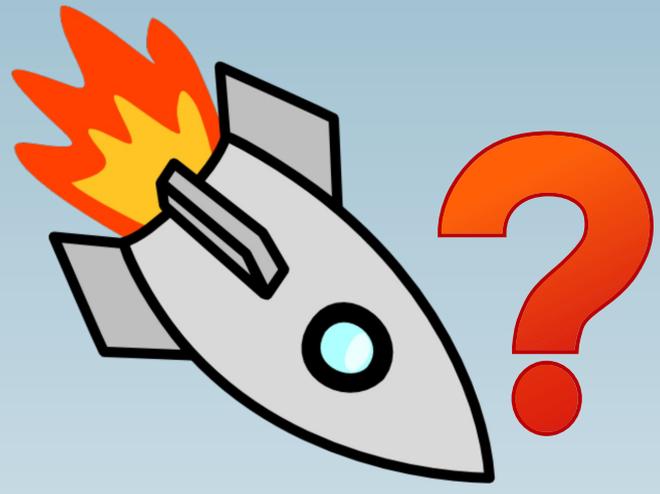
**in den
Bedingungen**

nicht

**berücksichtigt wurde
(Problem nicht
betrachtet)**

**abgedeckt ist
(Risiko nicht
versichert)**

Backup wie Versicherung ...



Datenbank-Backup – welcher Zweck?

➔ **Tauglichkeit des Backup, Verifikation**

Echtdaten vollständig nutzen

Dem Datenschutz genügen

Material für die Entwicklung

Automatisierung

Weitere Einfluss-Faktoren

Individuelle Prüfung nötig:

- **Wie sind die Abläufe?**
(z.B. Papier, Telefon oder Online-Änderungen?)
- **Welche technischen, kommerziellen und juristischen Randbedingungen?**
(z.B. Replikation von Daten und Log?)

Setzen Anforderungen an Backup

(z.B. Point-in-Time-Recovery gefordert ja/nein?
Dafür separate Log-Sicherung nötig ja/nein?)

Backup-Strategie

Ergibt sich aus technischen Möglichkeiten, Risiko-Annahmen, anderen Einfluss-Faktoren und Kosten:

- **Art, Umfang, Häufigkeit
(physisch vs logisch; mit/ohne Log; ...)**
- **Benutztes Tool**
- **Ablage / Speicherung**
- **Aufbewahrungsdauer**

Zweck des DB-Backup

Datenverlust verhindern

- ... durch Restore (Recovery)
- ... gemäß Anforderungen
- ... bei allen (betrachteten) Risiken

**Backup hat nur dann seinen Zweck erfüllt,
wenn Restore (Recovery) erfolgreich ist
=> Prüfkriterium des Backup**

Risiken für Recovery z.B. ...

- **Backup nicht für Recovery tauglich**
- **Defekter Backup erzeugt
(HW, SW, Netz, Platte voll, ...)**
- **Backup nach Ablage defekt / unerreichbar**
- **Recovery-Prozedur fehlerhaft / unpassend**
- **Recovery scheitert an System-Zustand
(HW, SW, Netz, Platte voll, ...)**

Verifikation – wie?

- **Nur erfolgreiche Recovery beweist grundsätzliche Tauglichkeit der Techniken von Backup und Recovery**
- **Kontrolle muss Routine-Aktion sein**
- **Kein Limit nach oben:
Prüfung von Schema, Satz-Zahlen, Inhalten,
Benutzer, Privilegien, ... ist möglich**

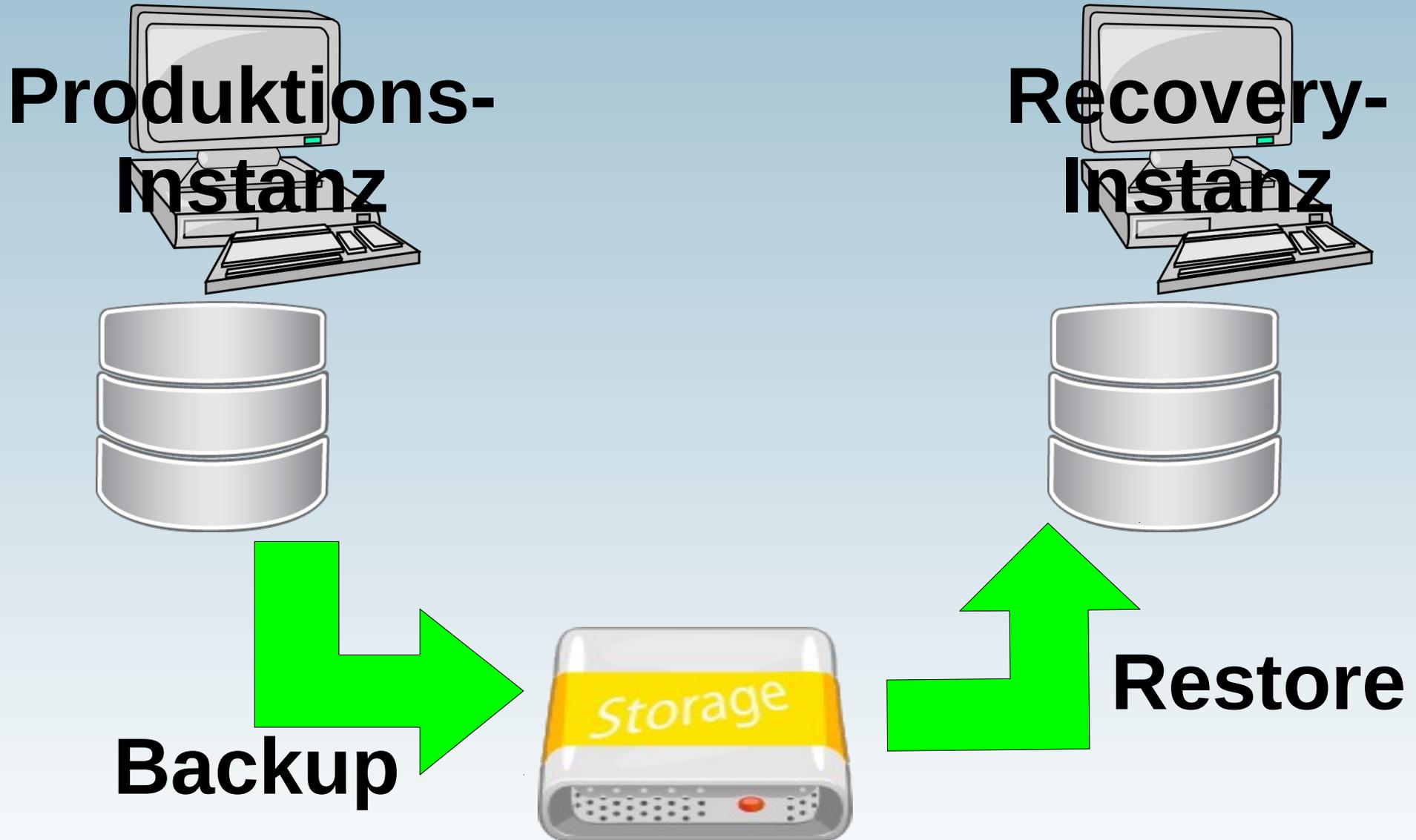
Verifikation – wo?

- **Produktiv-Backup enthält sensible Daten**
- **Produktive DB darf nicht betroffen sein**

Nötig:

**Eigenes Recovery-System in Produktion,
das für jede Produktiv-DB hinreichend groß ist**

Separate Recovery-Instanz



Datenbank-Backup – welcher Zweck?

Tauglichkeit des Backup, Verifikation

➔ **Echtdaten vollständig nutzen**

Dem Datenschutz genügen

Material für die Entwicklung

Automatisierung

Zustand nach Recovery-Test

- **Produktiv-DB**
 - mit Echt-Daten (aktuell)
 - im laufenden Betrieb (Zugriffe, Last, laufende Änderungen)
- **Recovery-DB**
 - mit Echt-Daten (vom Backup-Zeitpunkt)
 - auf separatem System (Leerlauf, stabil)

Kopie der Echt-Daten

- ... unterliegt dem Datenschutz
- ... taugt für Statistik, Abrechnung, ...
- ... erlaubt zuverlässige Messung von z.B. Schema-Änderungen („create index“, ...) (ggfs. HW-Unterschiede beachten)

Nutzungs-Beispiele (1)

- Abrechnung der letzten Periode (Woche, Monat, ...)
- Statistik / Report, besonders wenn ohne Index-Unterstützung
- Kontrolle auf Fehlerfreiheit
`mysqldump > /dev/null` für vollständiges Lesen
- ...

Nutzungs-Beispiele (2)

- **Export in anderem Format, z.B.**
 - **CSV für Spreadsheet**
 - **Mysqldump (Klartext) für Transfer**
- **...**

**Jede nur-lesende nicht-sofortige
Nutzung der Echtdaten
kann auf den Recovery-Server
verlagert werden.**

Datenbank-Backup – welcher Zweck?

Tauglichkeit des Backup, Verifikation

Echtdaten vollständig nutzen

➔ **Dem Datenschutz genügen**

Material für die Entwicklung

Automatisierung

Stichworte zum Datenschutz

- **Personenbezogene Daten**
z.B. Name, Adresse, Telefon, E-Mail, Konto
- **Zweckbindung der Daten**
- **Datensparsamkeit**
- **Fristen für Aufbewahrung / Löschung**
- **Abstimmung mit Datenschutz-Beauftragtem**
- **Im Zweifel: juristischen Rat holen**

Anonymisieren

Restore-Instanz hat SQL verfügbar:

- **UPDATE kunde**
SET name = 'Mustermann',
vorname = 'Max'
WHERE anrede = 'Herr';

- **UPDATE kontakte**
SET mail = concat(md5(mail),
'@provider.tld');

Eigeninitiative

- **Pessimistische Annahme:
Die anderen vergessen den Datenschutz!**
- **Ausländische Kollegen ...**
- **Als DBA selbst das Schema prüfen,
bei Zweifeln nachfragen**
- **Datenschutz-Belehrung beachten**

Hoffentlich überflüssig

- **Langfristiges (Firmen-) Interesse**
- **vgl. aktuelle Skandale und Bußgelder**
- **Zivilcourage !**

Datenbank-Backup – welcher Zweck?

Tauglichkeit des Backup, Verifikation

Echtdaten vollständig nutzen

Dem Datenschutz genügen

➔ **Material für die Entwicklung**

Automatisierung

Entwickler-Wünsche

- **Analyse der Feature-Nutzung**
- **Test mit Echt-Daten**
Volumen, Werte-Verteilung
- **Daten-Volumen für Tuning**
„Explain“ mit/ohne Index
- **Schema-Änderung testen**
„Create Index“, „Alter Table“

Probleme / Konflikte

- **Datenschutz-Recht:**
 - Gesetzliche Verpflichtung
 - Zweckbindung
 - Löschung
- **Datenschutz-Erklärung:**
 - Eigene Zusagen an Nutzer
- **Datenschutz-Belehrung:**
 - Vertraulichkeit, ...

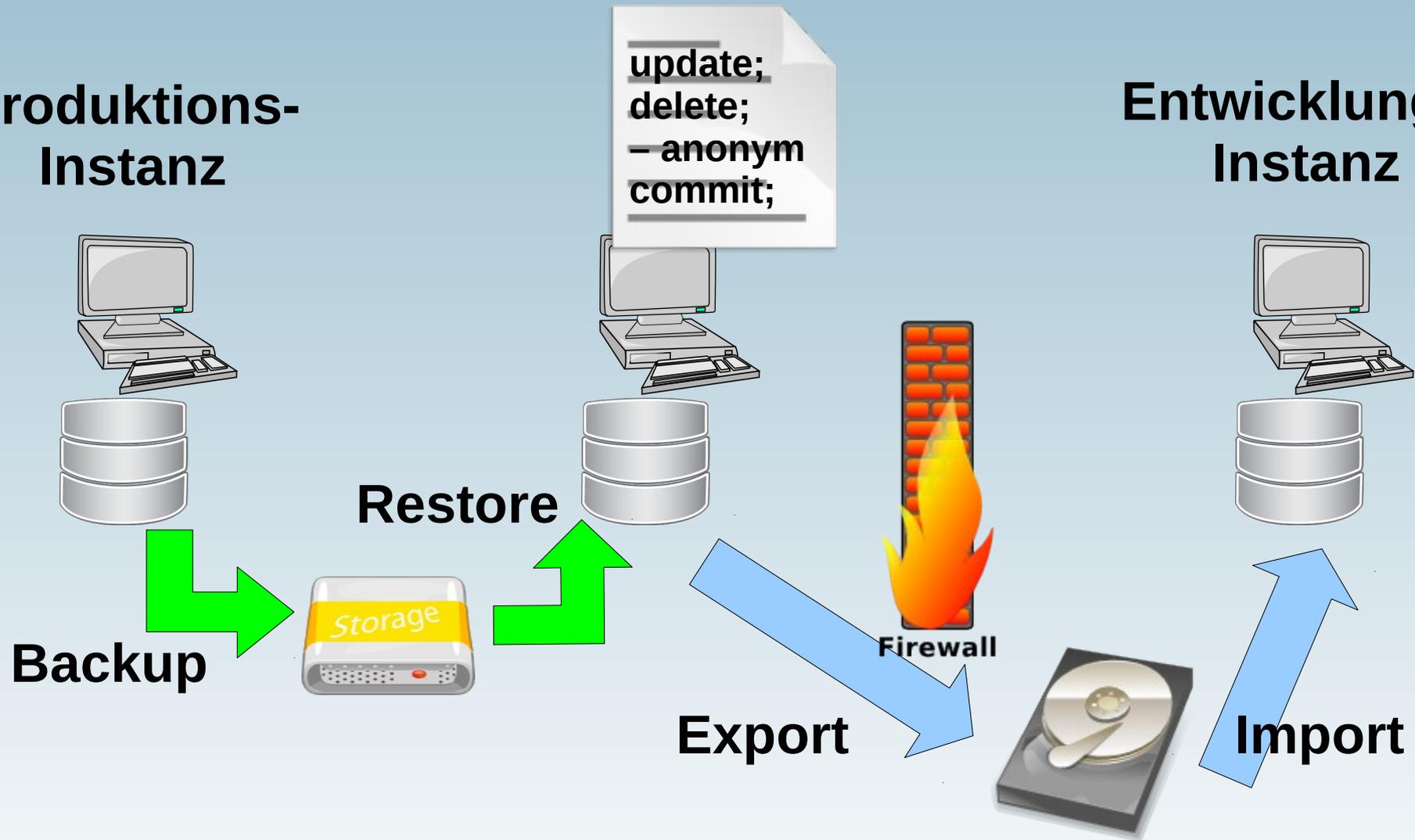
Ausweg / Lösung

- **In Recovery-Instanz**
 - löschen
 - anonymisieren
 - nur selektiv exportieren
- **Export ablehnen**
- **Entwickler-Skript annehmen
und als DBA kontrolliert auf Recovery-Instanz
ausführen**

Export zur Entwicklung

**Produktions-
Instanz**

**Entwicklungs-
Instanz**



Datenbank-Backup – welcher Zweck?

Tauglichkeit des Backup, Verifikation

Echtdaten vollständig nutzen

Dem Datenschutz genügen

Material für die Entwicklung

➔ **Automatisierung**

Vorbemerkung

- **Beispiele / Code für MySQL, bitte selbst an Firmen-Regeln anpassen und für anderes DBMS übertragen**
- **\$CONNECT = Verbindungs-Information
-u USER -pPASSWORD -h HOST**
- **\$DB = ausgewählte Datenbank (Schema)**

Automatisierung: Prinzipien

- **Keine manuelle Ausführung von Routine-Aufgaben**
- **Erfolgs-Kontrolle im Skript enthalten, manuell nur Stichproben**
- **Mail-Alarm bei Fehler**
- **Keine manuelle Konfiguration / Anpassung, Skript erstellt Listen selbst**

Automatisierung: Teile

- **Backup (lokale MySQL-Instanz)**
- **Purge Log**
- **Restore (Sicherung vorgegeben)**
- **Restore-Wrapper (automatisierte Auswahl)**
- **DBen/Schemata im aktuellen Restore**
- **User/Passwort für DB/Schema**
- **Anonymisierungs-Skript für DB/Schema**

Skript: Backup (1)

- **Sichert DB-Instanz gemäß Backup-Strategie**
- **Generiert Namen mit Datum/Uhrzeit**
- **Löscht überflüssige alte Backups**
- **Aufruf durch cron**

- **Separat: Monitoring für Speicherplatz**

Skript: Backup (2)

- **Hilfreich: Tabelle „backup_history“ mit Timestamp und Exit-Code**
 - **„MySQL Enterprise Backup“ führt sie**
 - **Auf Slave repliziert?**
MEB 3.9 ja, 3.10 nein
Oracle SR 3-9497158271
- **Welche Instanz(en) sichern bei Replikation?**

Skript: Purge Log (1)

Bis zum vorletzten erfolgreichen Backup löschen:

```
TIMESTAMP=`echo  
    "select start_time from  
        mysql.backup_history  
        where exit_state = 'SUCCESS'  
        order by start_time desc limit 1, 1 ;"  
| mysql $CONNECT | tail -n1`  
  
echo  
    "PURGE MASTER LOGS BEFORE '$TIMESTAMP' ;"  
| mysql $CONNECT
```

Skript: Purge Log (2)

Ohne Tabelle – fest 26 Stunden zurück:

```
TIMESTAMP=`echo`  
  "select subtime( now(), '26:00:00')"  
| mysql $CONNECT | tail -n1`
```

Skript: Restore

- **Name der Sicherung wird als Parameter übergeben**
- **Rückfrage, falls nicht auf Restore-Maschine**
- **Löscht alle vorhandenen DB-Inhalte**
- **Lädt Sicherung**
- **Startet DB-Server**

Skript: Restore-Wrapper (1)

- **Sorgt dafür, dass die Sicherungen aller DB-Instanzen der Test-Recovery unterzogen werden**
- **Läuft nur auf Restore-Host**
- **Optionaler Parameter: Hostname**
 - **Gegeben: Name des letzten Backup bestimmen**
 - **Sonst: alle Instanzen reihum**
- **Ruft Restore-Skript mit Sicherungsname**

Skript: Restore-Wrapper (2)

Round-Robin über alle Hostnamen:

```

HOSTLIST=( `ls -d ... |
            grep -iv ...` )
NUMHOSTS=${#HOSTLIST[@]}      # count
DAY=`date +%j`                # day of year
INDEX=`expr $DAY % $NUMHOSTS`
BACKUPDIR=${HOSTLIST[$INDEX]}

```

```

TIMESTAMP=`ls $BACKUPDIR |
            grep '^20[1-9][0-9]' | tail -1`

```

Code: Welche DB/Schema?

```
DBLIST=`cd /DISK ; find * -type d`
for DB in $DBLIST
do
    if [ "$DB" = "information_schema"
        -o "$DB" = "mysql"
        -o "$DB" = "performance_schema" ]
    then
        echo "Ignoring database $DB"
        continue
    fi
    ... # nächste Folie
done
```

Code: User/Passwort zu DB

Annahme: Benutzer existiert schon in der DB und hat die nötigen Privilegien für Anonymisierung

Sonst: Nach Restore erzeugen

```
DB_S=`echo $DB |  
    tr '[:upper:]' '[:lower:]'`  
# 9 Zeichen von DB für User(16)  
DB_9=`echo ${DB_S} |  
    sed 's/\(.....\) .*/\1/'`  
DB_USR="${DB_9}_anonym"  
DB_PWD="${DB_S}#geheim#"
```

Code: DB behandeln?

DB ohne Anonymisierungs-Nutzer wird nicht behandelt

```
mysql -u $DB_USR -p$DB_PWD -e 'show grants;'
RC=$?
if [ $RC -ne 0 ]
then
    echo "'mysql -u $DB_USR -p$DB_PWD'
        scheitert mit Code $RC"
    echo "Kein Check / Export für diese DB."
    continue
fi
```

Code: Skript suchen

```
for ACTION in pruef anonym export ; do
    BASE=/pfad/zu/script-${ACTION}-${DB_S}
    SCRIPT=${BASE}.sh
    if [ -x $SCRIPT ] ; then
        bash $SCRIPT -U $DB_USR -P $DB_PWD -N $DB_S
        RC=$?
        echo "$SCRIPT terminated with exit code $RC"
    fi
    SCRIPT=${BASE}.sql
    if [ -r $SCRIPT ] ; then
        mysql -u $DB_USR -p$DB_PWD $DB_S < $SCRIPT
        RC=$?
        echo "$SCRIPT terminated with exit code $RC"
    fi
done
```

Skript: Anonymisieren (1)

Überflüssige Tabellen leeren:

```
echo "use information_schema ;  
      select TABLE_NAME from TABLES  
      where TABLE_SCHEMA = '$DB' and  
            TABLE_TYPE = 'BASE TABLE' and  
            TABLE_NAME not in ('t1', 't2', 't3') ;" | \  
mysql $CONNECT --skip-column-names | \  
while read TAB  
do  
      echo "truncate table $TAB ;"  
done | mysql $CONNECT $DB
```

Skript: Anonymisieren (2)

Personenbezogene Daten:

```
mysql $CONNECT $DB <<'eof'  
UPDATE t1 SET  
    payment = 'xxx', birth = '1970-01-01',  
    fname = 'Hans', lname = 'Mustermann',  
    email = 'mustermail@muster.comp',  
    ... ;  
  
UPDATE t2 SET  
    request = NULL, response = NULL;  
eof
```



Code: „select count(*)“

```
# Get a list of all tables, feed it into a loop
# that generates "select count(*)" statements,
# pipe these to another client call for execution

mysql $CONNECT --silent --skip-column-names
-e "select TABLE_NAME from
    information_schema.TABLES
    where TABLE_SCHEMA = '$DB' and
    TABLE_TYPE = 'BASE TABLE' order by 1" | \
while read TN
do
    echo "select count(*) as '$TN' from $DB.$TN ;"
done | mysql $CONNECT --table $DB
```

Code: „show create table“

```
# Get a list of all tables, feed it into a loop
# that generates "show create table" statements,
# pipe these to another client call for execution

echo 'show tables ;' | \
mysql $CONNECT $DB | tail -n +2 | sort | \
while read TAB
do
    echo "show create table $TAB \G"
done | mysql $CONNECT $DB
```

Q & A



www.fromdual.com



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