



Die Spider Storage Engine für MySQL

Skalierung über Instanzen hinweg

Ralf Gebhardt
Principal Sales Engineer



MariaDB Corporation

- Gegründet vom MySQL Kernteam, wie Michael “Monty” Widenius und David Axmark.
- Oktober 2014 wurde die Firma von SkySQL in MariaDB Corporation umbenannt.
- Entwicklungs-Unterstützung der MariaDB Foundation
- Innovative Produkte werten MariaDB auf.
- Enterprise Services wie 24/7 Support for MariaDB and MySQL.
- Über 400 Jahre an MySQL Erfahrung bei MariaDB.
- Open-Source- und Cloud-Management-Team mit Hauptsitz in Espoo, Finland.

The screenshot shows the MariaDB website header with navigation links: Products, Services, Resources, News & Events, About, Blog, Login, and Sign Up. The main content area features a prominent announcement: "MariaDB is a Gartner 'Leader'" followed by "Magic Quadrant for Operational Database Management Systems 2014 Report". Below this, a text box states: "MariaDB is regarded as a 'Leader' in the operational database management system market by the world's leading information technology research and advisory company, Gartner." A "Read Report" button is visible. To the right, a Gartner logo and a small thumbnail of the report are shown. Below the announcement, a section titled "Three top reasons to use MariaDB Enterprise" lists: "Famously Open", "Global Expertise", and "Certified Binaries".



Sharding

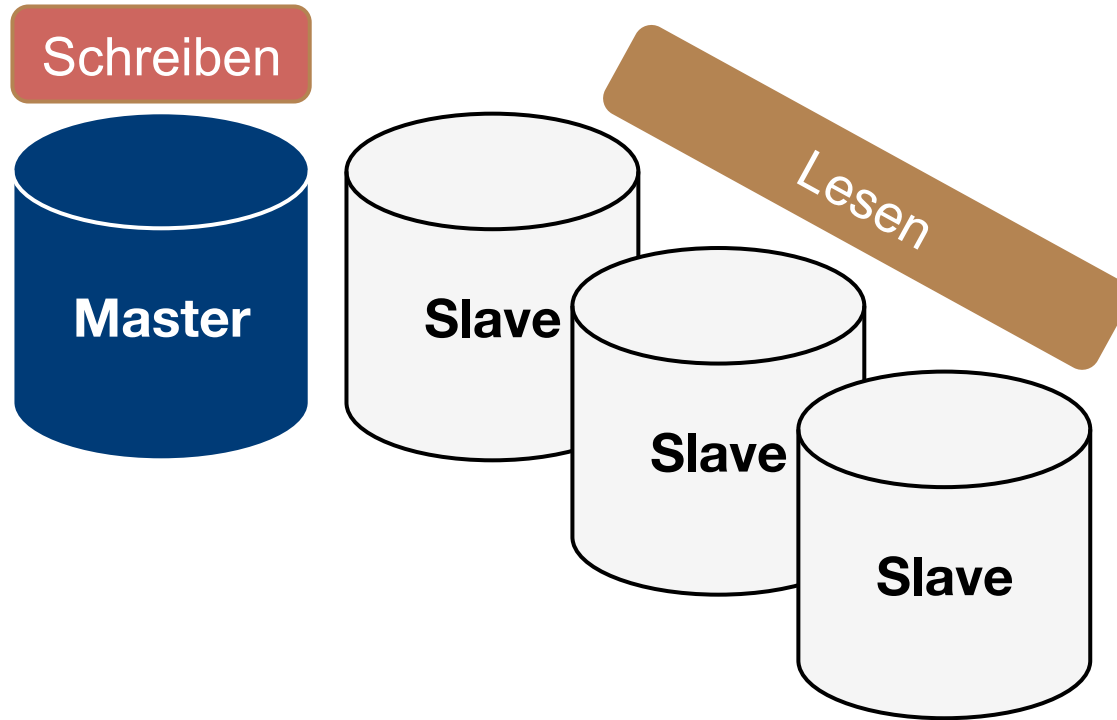


Sharding – Wann?

- Große Datenmengen
 - Engpässe bezüglich I/O- und CPU-Last einer Server-Instanz
 - Lange Antwortzeiten von Abfragen
 - Auswirkungen auf Indizierung, Statistiken, Tabellen-Verwaltung, ...
- Wenn Replikation nicht die Antwort ist
- Wenn Partitionierung innerhalb einer Instanz nicht ausreicht



Replikation oder Sharding?



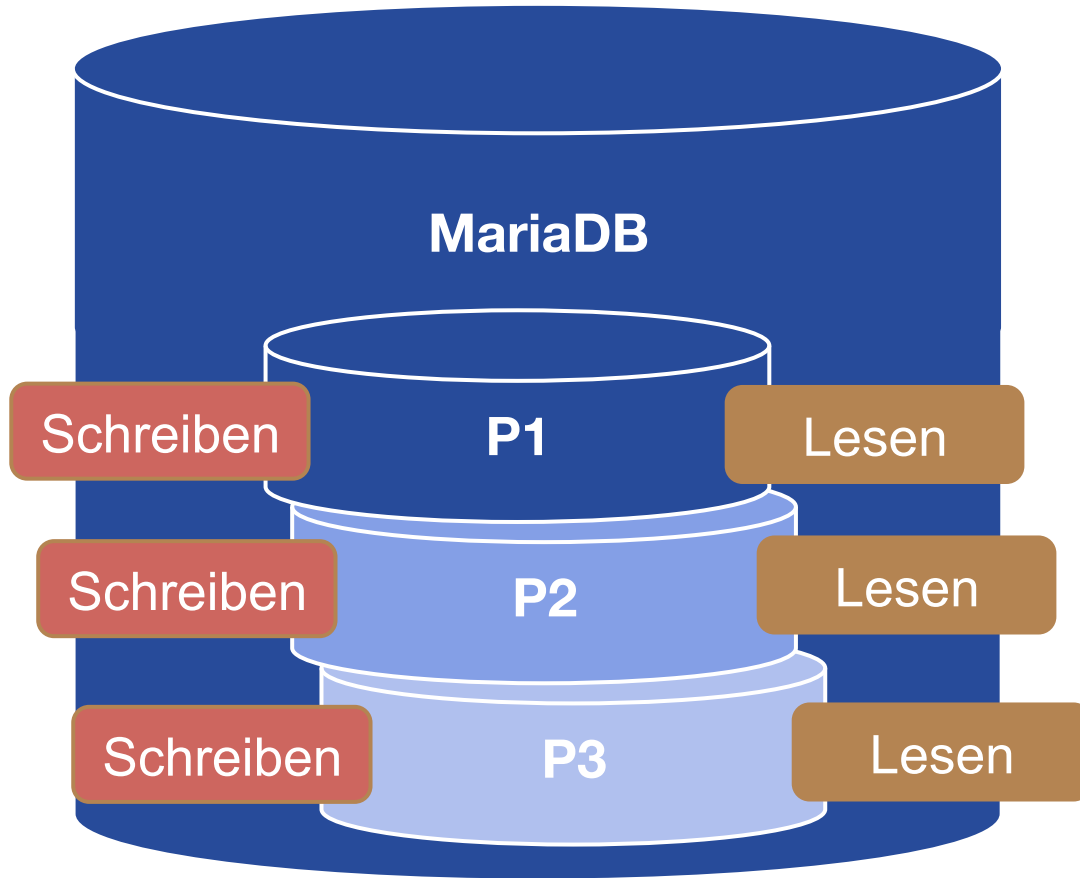


Replikation oder Sharding?

- Master/Slave-Replikation
 - Lese-Skalierung bei großer Anzahl Connections und Abfragen
 - Keine Lösung bei hohen Antwortzeiten einzelner Abfragen durch große Datenmengen
 - Keine Skalierung für Schreib-Last

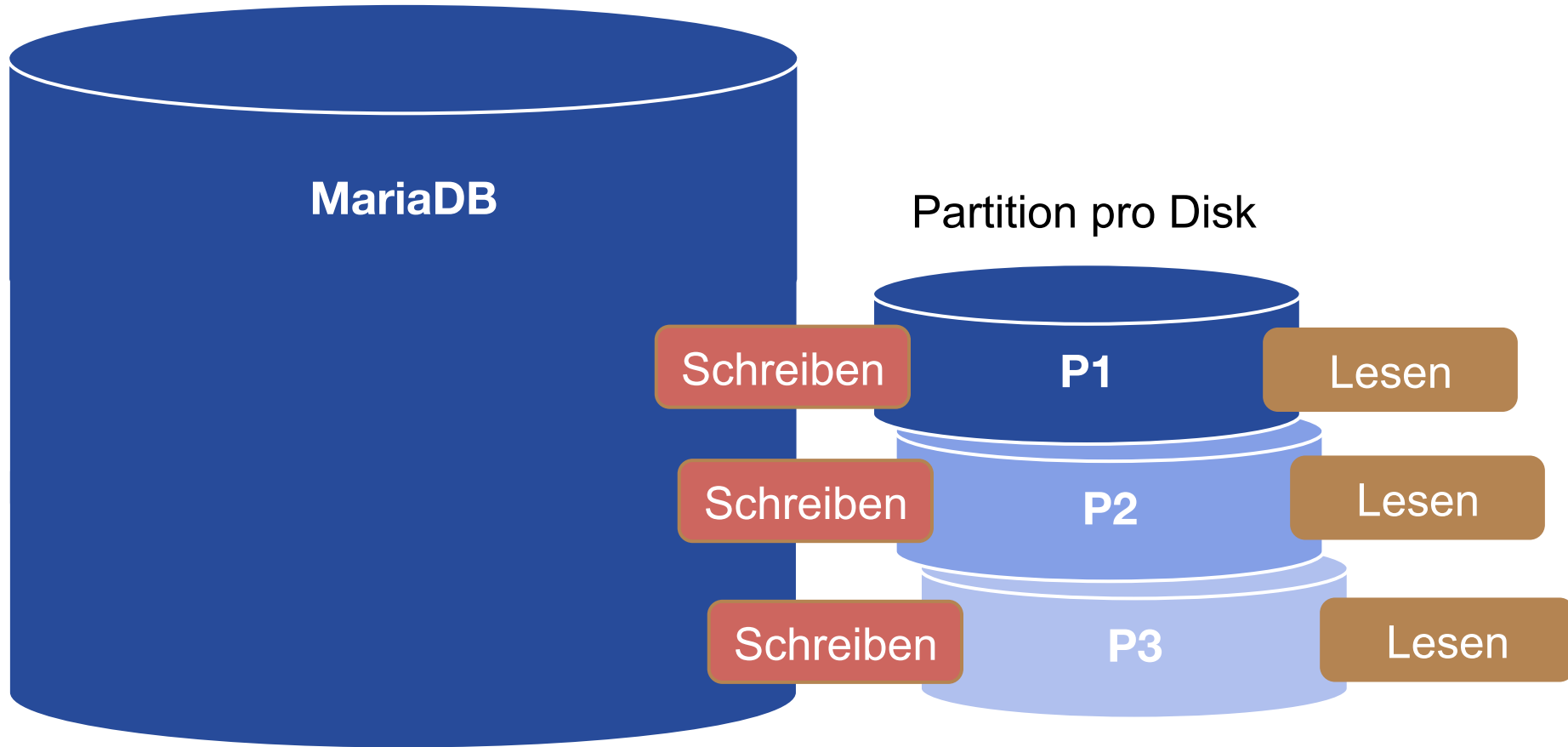


Partitionierung oder Sharding?





Partitionierung oder Sharding?





Partitionierung oder Sharding?

- Partitionierung ermöglicht
 - Die Reduzierung des Datenbestands für gängige Abfragen, wenn die Partitionierungsregel geeignet gewählt werden kann
 - Trennung zwischen Archivdaten und „Working-Set“
 - Die Verteilung von I/O-Last auf mehrere Disks
- Ressourcen einer Instanz müssen geteilt werden (CPU, RAM, Kern-Prozesse, ...)

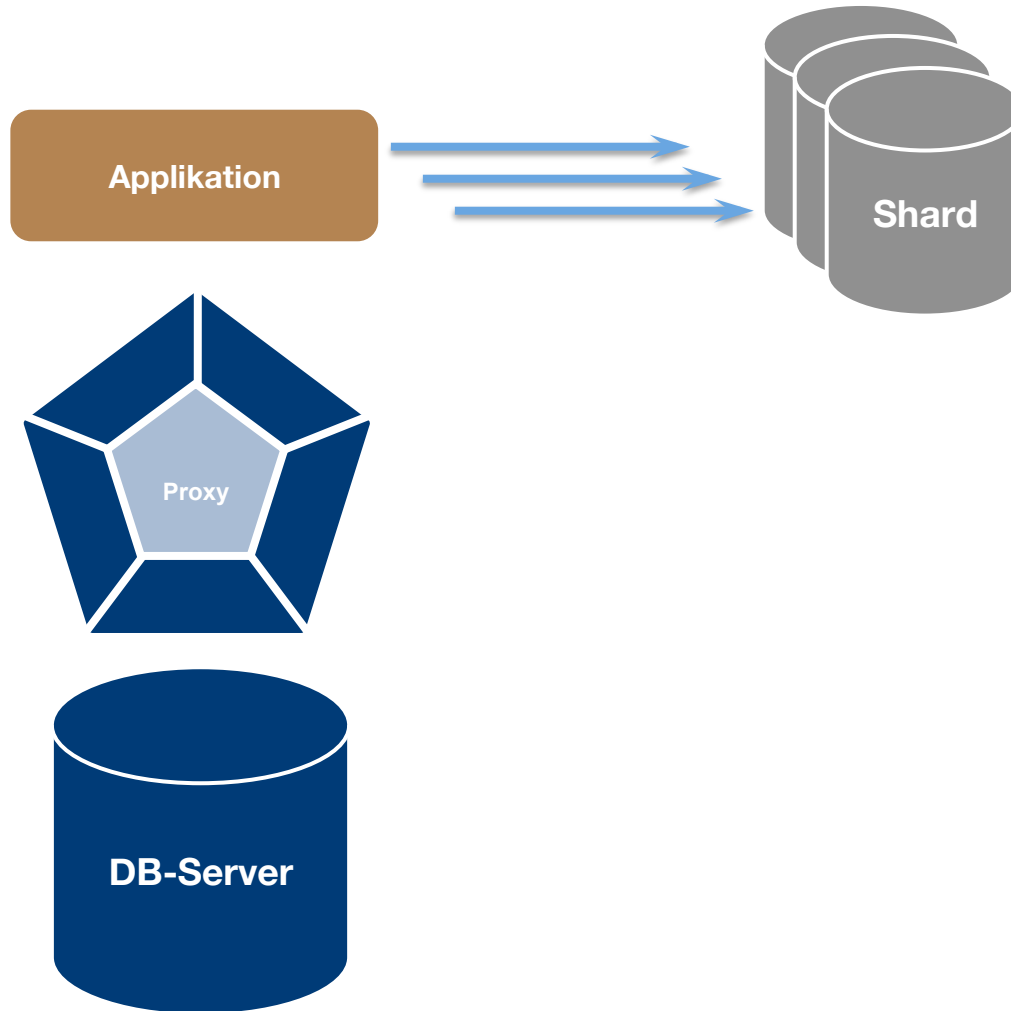


Sharding – Wie?

- Sharding ist Datenbankpartitionierung über mehrere Serverinstanzen
- Implementierung von Sharding über
 - Anwendungslogik
 - Konnektoren
 - Proxies (MySQL Proxy, MySQL Fabric, MariaDB MaxScale)
 - Spider Storage Engine
 - ...

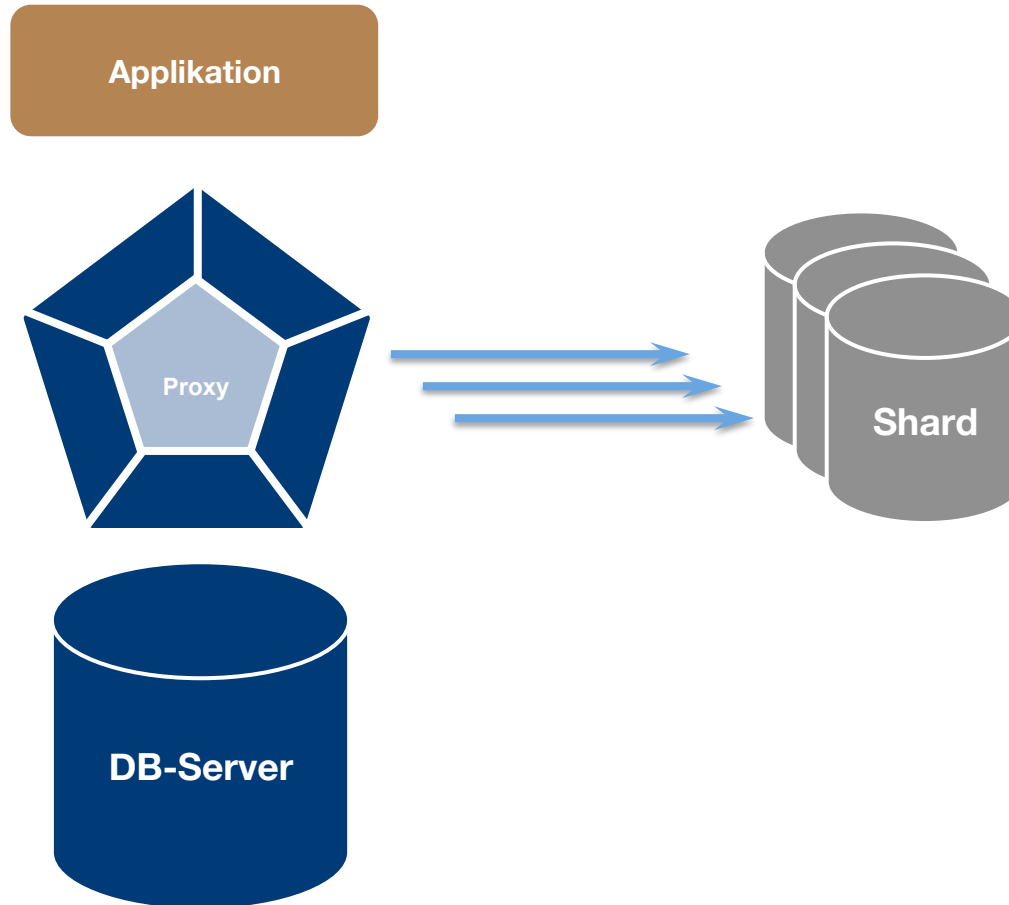


Sharding – Wie?



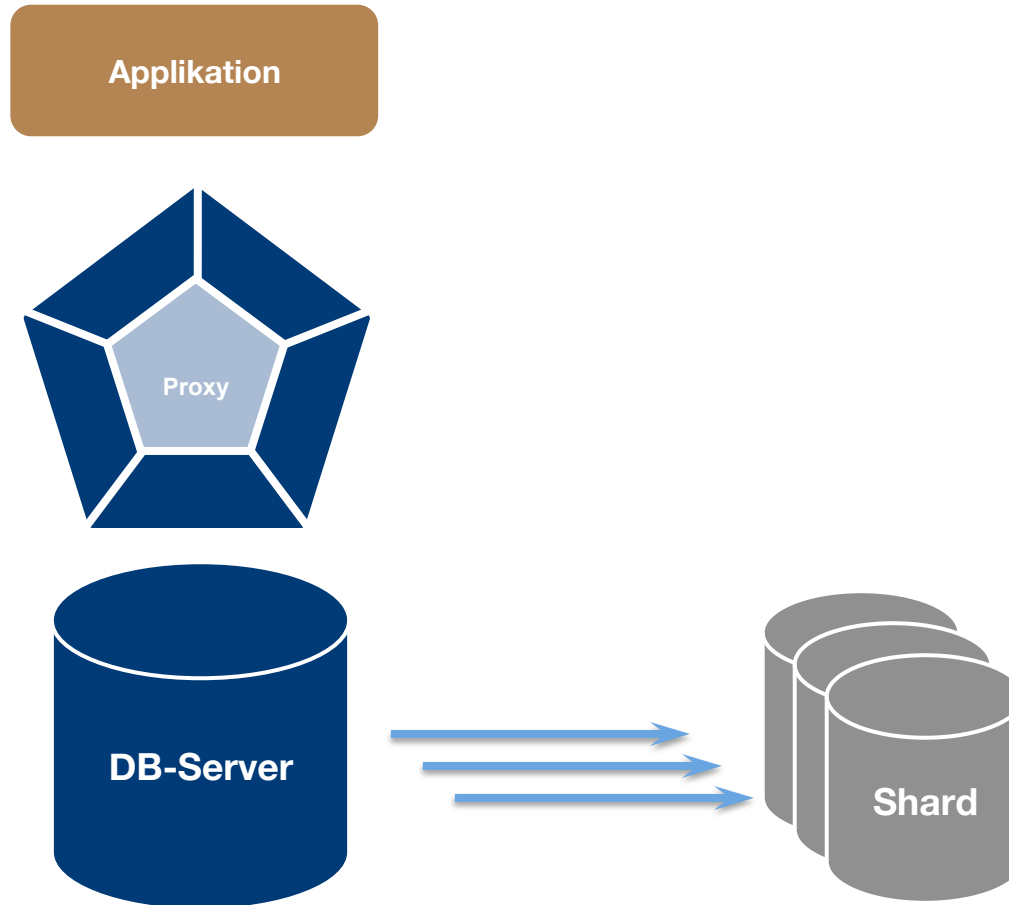


Sharding – Wie?





Sharding – Wie?

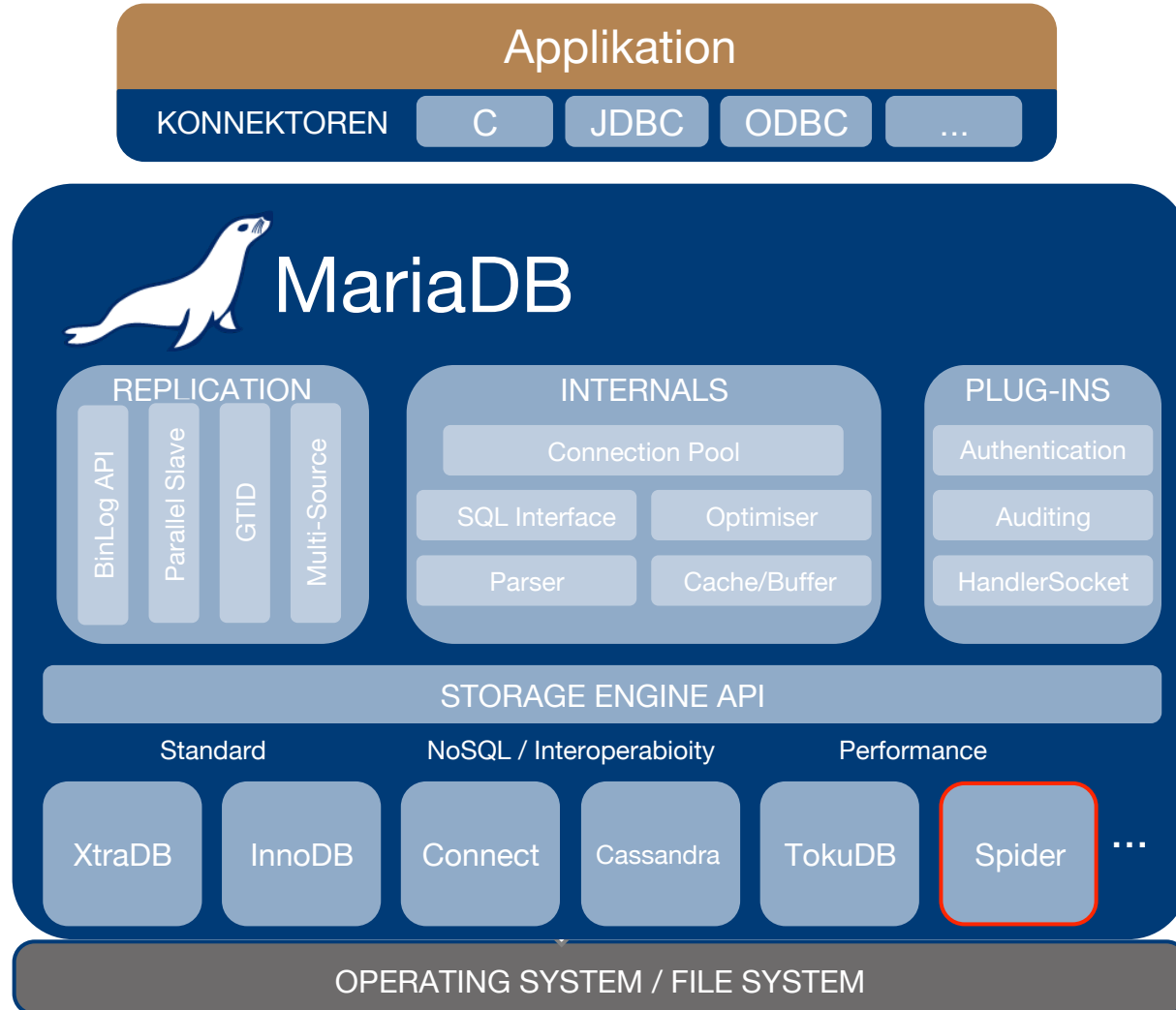




Spider Storage Engine



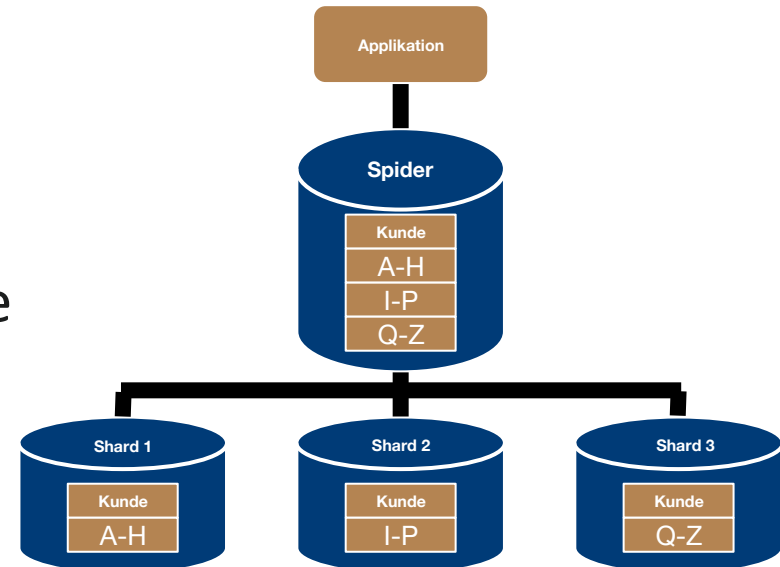
Storage Engine Architektur





Die Spider Storage Engine

- Entwickelt von Kentoku Shiba
- Storage Engine partitioniert große Tabellen über mehrere Datenbankserver-Instanzen.
- Basiert auf Partitionierung mit integriertem Sharding
- Virtuelle Sicht auf Tabellen verschiedener Instanzen
- Unterstützt XA Transaktionen
- Transaktionale Storage Engine
- Ermöglicht Scale-Out in Kombination mit HA



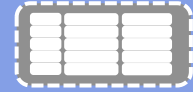


Grundidee der Spider Engine

- Applikation mit Verbindung zum Spider Proxy-Knoten
- `CREATE TABLE spider (...) ENGINE=SPIDER ...`
 - „Spider-Proxy“ hält keine Daten
- `CREATE TABLE spider (...) ENGINE=INNODB ...`
 - Backend hält Daten

Applikation

Spider
Proxy-Knoten

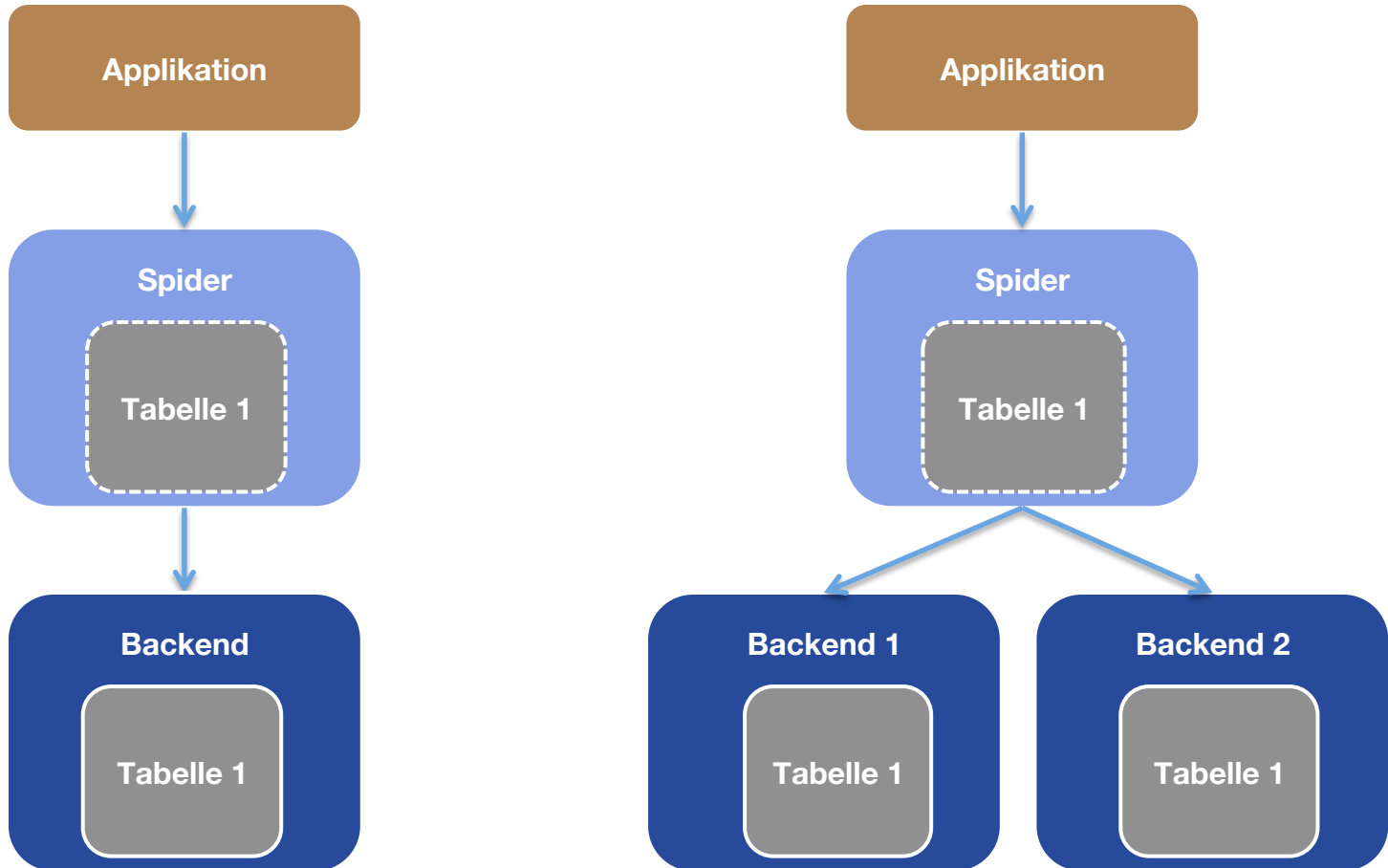


Backend-Knoten



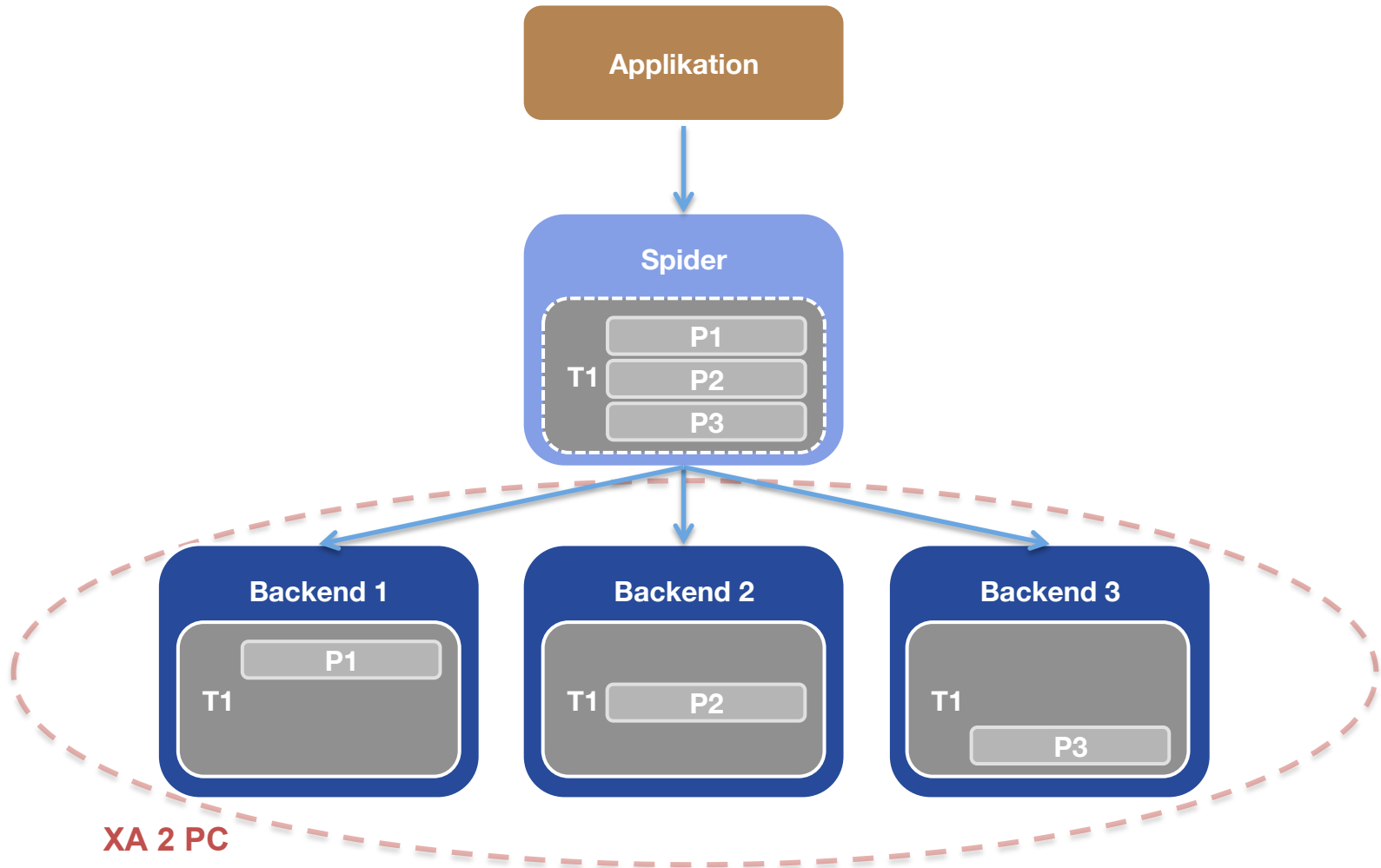


Spider als „Federation“ Ohne und mit HA



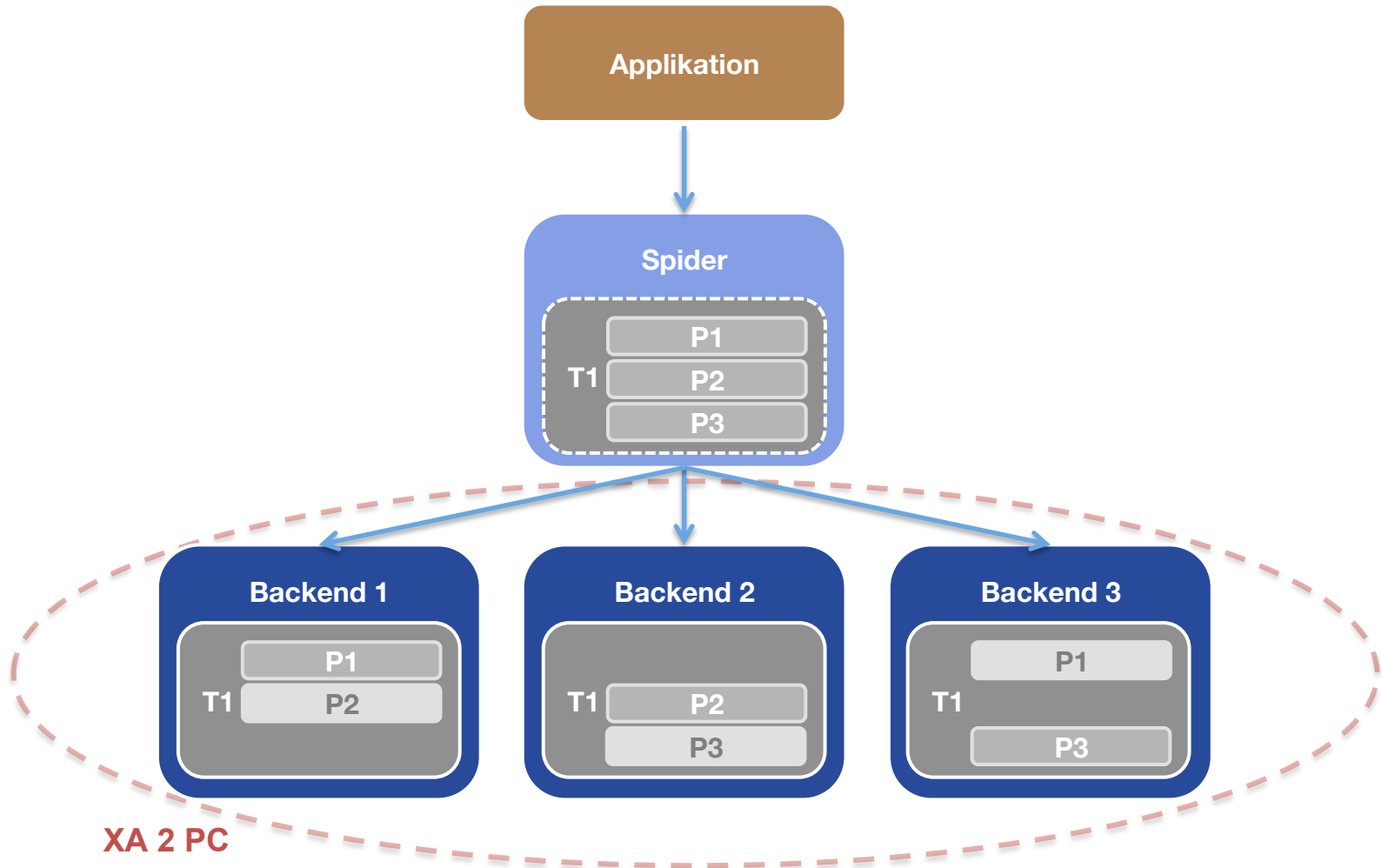


Sharding mit Spider





Sharding mit Spider und HA





Spider Storage Engine Installation

- Ab Version 10.0.4 in MariaDB enthalten
 - Spider 3.0
 - Spider 3.2.11 in MariaDB 10.0.14
- Spider mit MySQL Server
 - http://spiderformysql.com/download_spider.html
 - `INSTALL PLUGIN spider SONAME 'ha_spider.so';`



Spider Storage Engine Installation

- Zur Installation

```
mysql -uroot -p < /usr/share/mysql/install_spider.sql
```

- Spider sollte als aktive Storage Engine angezeigt werden

```
SELECT engine, support, transactions, xa FROM  
information_schema.engines;
```

engine	support	transactions	xa
SPIDER	YES	YES	YES
CSV	YES	NO	NO



Spider Storage Engine Installation

- Spider erstellt Tabellen in Schema „mysql“

```
MariaDB [mysql]> show tables like 'spider%';
+-----+
| Tables_in_mysql (spider%) |
+-----+
| spider_link_failed_log    |
| spider_link_mon_servers  |
| spider_tables             |
| spider_xa                 |
| spider_xa_failed_log     |
| spider_xa_member         |
+-----+
6 rows in set (0.00 sec)
```




Spider Storage Engine Installation

- 93 Spider System-Variablen werden hinzugefügt

```
MariaDB [mysql]> show global variables like 'spider%';
```

- 4 Spider Status-Werte werden hinzugefügt

```
MariaDB [mysql]> show global status like 'spider%';
```

- Weitere Spider Variablen für Tabellen stehen über CREATE TABLE zur Verfügung
 - In MariaDB über COMMENT
 - In MySQL über CONNECTION



Spider Storage Engine Installation

- Spider UDFs werden hinzugefügt
 - SPIDER_DIRECT_SQL
 - Ausführen von SQL direkt auf Backend-Server
 - SPIDER_BG_DIRECT_SQL
 - Ausführen von SQL direkt auf Backend-Server
 - SPIDER_COPY_TABLES
 - SPIDER_FLUSH_TABLE_MON_CACHE
 - Zurücksetzen von Spider-Monitoring-Information



Spider Storage Engine

Beispiel ohne Sharding

- Tabellendefinition Spider Proxy-Knoten

```
CREATE TABLE spiderfederation(id INT NOT NULL, code  
VARCHAR(10), PRIMARY KEY(id))  
ENGINE=SPIDER  
COMMENT 'host "192.168.56.21", user "backend", password  
"backend", port "3306"';
```

- Tabellendefinition Backend-Knoten

```
CREATE TABLE spiderfederation(id INT NOT NULL, code  
VARCHAR(10), PRIMARY KEY(id))  
ENGINE=INNODB;
```



Spider Storage Engine

Beispiel mit Sharding

- Tabellendefinition Spider Proxy-Knoten

```
CREATE TABLE sharding(id INT NOT NULL, code VARCHAR(10),  
PRIMARY KEY(id))  
ENGINE=SPIDER COMMENT='user "backend", password  
"backend", port "3306", table "sharding"  
PARTITION BY RANGE(id)  
(  
    PARTITION p1 VALUES LESS THAN (100000)  
    COMMENT 'host "192.168.56.21"',  
    PARTITION p2 VALUES LESS THAN (200000)  
    COMMENT 'host "192.168.56.22"',  
    PARTITION p3 VALUES LESS THAN MAXVALUE  
    COMMENT 'host "192.168.56.23" '  
);
```



Spider Storage Engine Beispiel mit Sharding

- Tabellendefinition der Backend-Knoten

```
CREATE TABLE sharding(  
  id INT NOT NULL,  
  code VARCHAR(10),  
  PRIMARY KEY(id)  
)ENGINE=INNODB;
```



Spider Storage Engine

Beispiel mit Sharding

- Insert Proxy

```
MariaDB [doag]> insert into sharding values (90002,"shard1"),  
(100100,"shard2"),(200050,"shard3");  
Query OK, 3 rows affected (0.04 sec)  
Records: 3  Duplicates: 0  Warnings: 0
```

- Shard 1

```
MariaDB [doag]> select * from sharding;  
+-----+-----+  
| id      | code    |  
+-----+-----+  
| 90001   | shard1  |  
| 90002   | shard1  |  
+-----+-----+  
2 rows in set (0.00 sec)
```



Spider Storage Engine

Beispiel mit Sharding

- Shard 2

```
MariaDB [doag]> select * from sharding;
+-----+-----+
| id      | code   |
+-----+-----+
| 100050  | shard2 |
| 100100  | shard2 |
+-----+-----+
2 rows in set (0.00 sec)
```

- Shard 3

```
MariaDB [doag]> select * from sharding;
+-----+-----+
| id      | code   |
+-----+-----+
| 200000  | shard3 |
| 200050  | shard3 |
+-----+-----+
```



Spider Storage Engine

Beispiel mit Sharding

- ACHTUNG – Kein automatisches Rollback!

```
MariaDB [doag]> begin;
Query OK, 0 rows affected (0.00 sec)

MariaDB [doag]> insert into sharding values
(90003,"shard1");
Query OK, 1 row affected (0.01 sec)

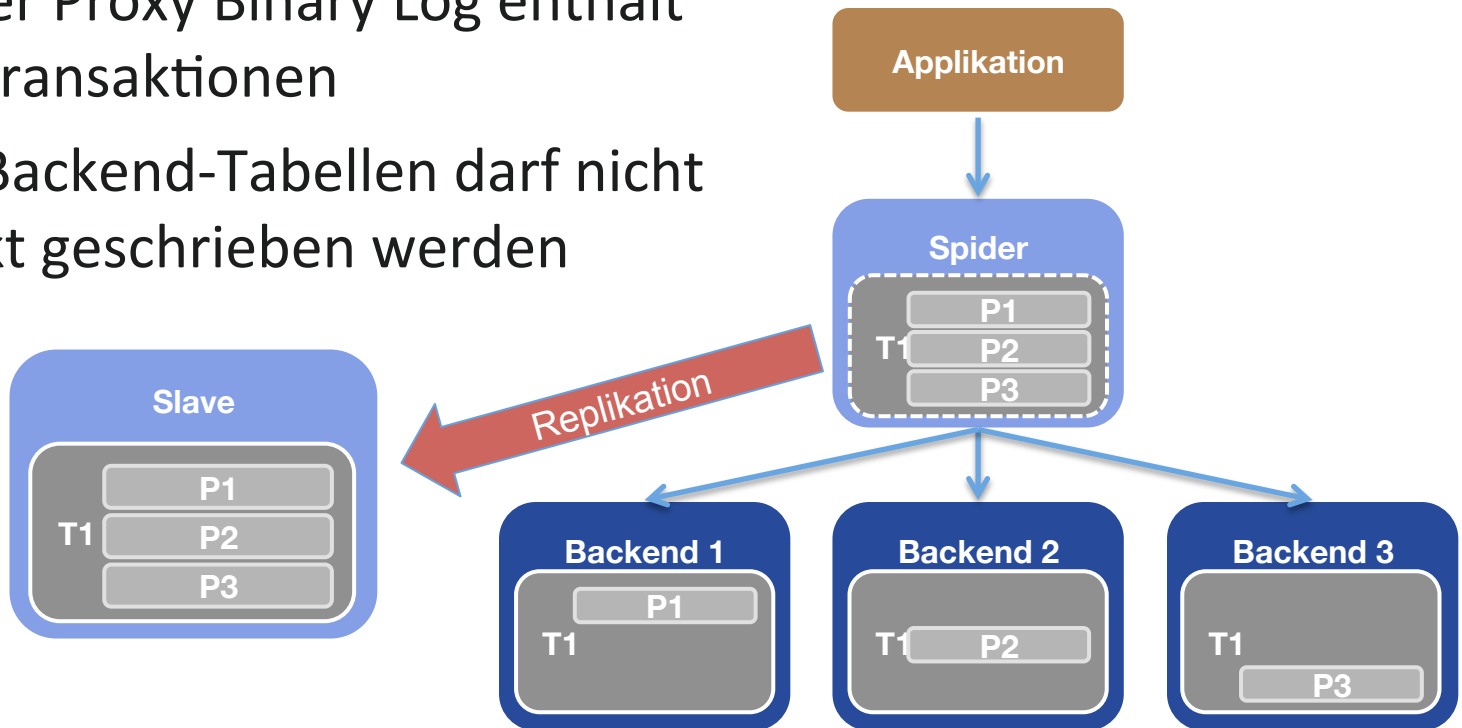
MariaDB [doag]> insert into sharding values
(100101,"shard2");
Query OK, 1 row affected (0.00 sec)

MariaDB [doag]> insert into sharding values
(200051,"shard3");
ERROR 1429 (HY000): Unable to connect to foreign data
source: 192.168.56.23
MariaDB [doag]> commit;
Query OK, 0 rows affected (0.01 sec)
```




Backup und Reporting

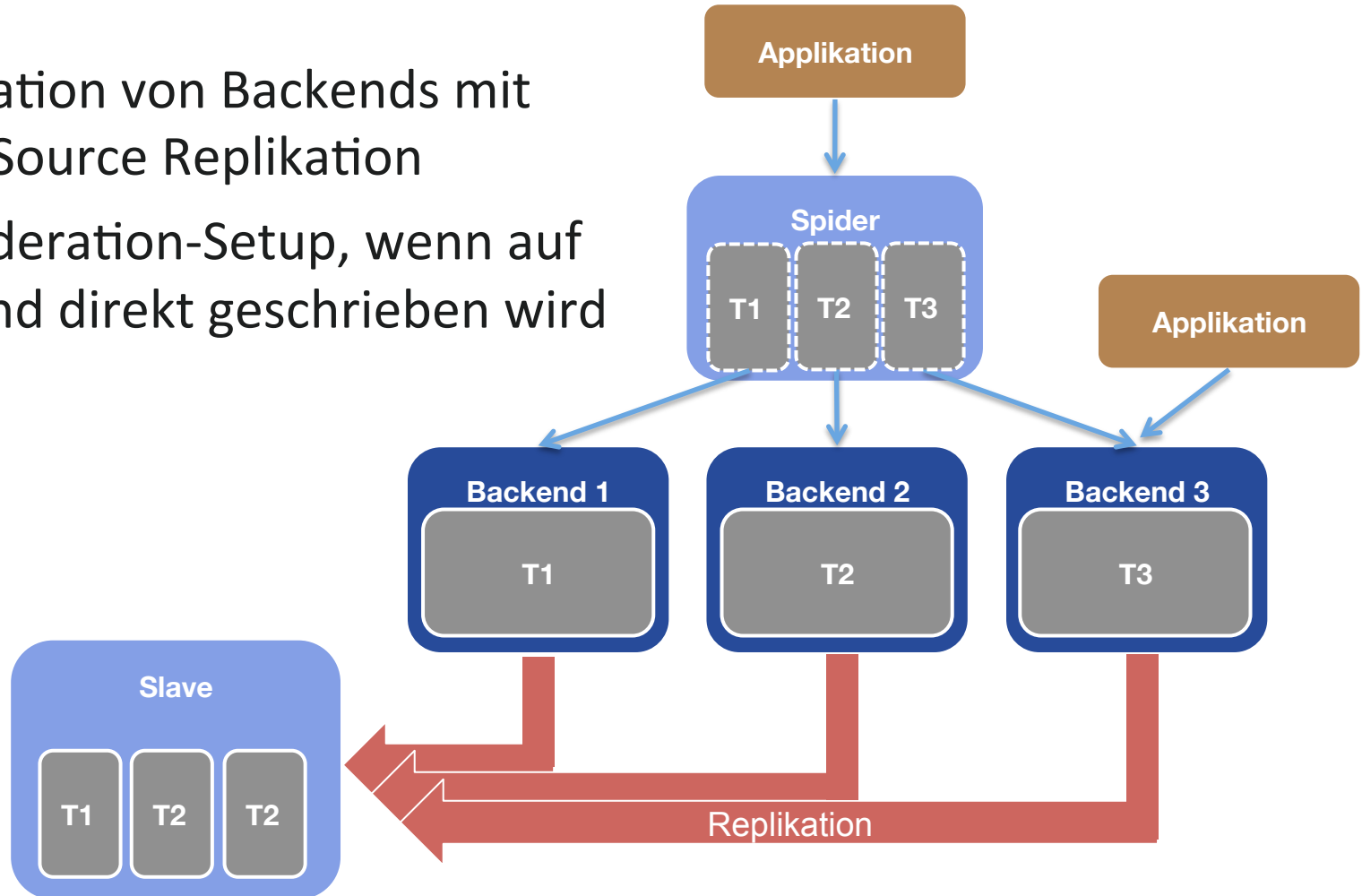
- Replikation von Spider Proxy zu Slave mit z.B. InnoDB
- Spider Proxy Binary Log enthält die Transaktionen
- Auf Backend-Tabellen darf nicht direkt geschrieben werden





Backup und Reporting

- Replikation von Backends mit Multi-Source Replikation
- Bei Federation-Setup, wenn auf Backend direkt geschrieben wird





Spider Funktionen

- Komplette Liste unter <https://mariadb.com/kb/en/mariadb/documentation/storage-engines/spider/spider-feature-matrix/>
- Clustering und Hochverfügbarkeit
 - Commit und Rollback auf mehreren Backends
 - Multiplexing zu Repliken mit 2PC
 - Split-Brain-Resolution basierend auf Quorum



Spider Funktionen

- Performance
 - Index Condition Pushdown (ab MariaDB 10 RC)
 - Engine Condition Pushdown für Federated
 - Engine Condition Pushdown für Shards (MariaDB 10 RC)
 - Batched Key Access
 - Support für Handler Socket
 - Map Reduced für ORDER BY ... LIMIT



Zu Beachten

- DDL Statements werden nicht synchronisiert
- Effektivität von Sharding entspricht der von Partitionierung
 - Sub-Partitionen für Backends möglich
- Pro-Instanz Funktionalität beachten
 - Query Cache sollte deaktiviert werden
 - Log-Files pro Instanz
 - Audit-Plugin – u.U. zentrales Syslog nutzen
 - Benutzer-Verwaltung - Authentication Plugin?
- **Spider Storage Engine noch BETA**



Für mehr Information ...

- <https://mariadb.com/kb/en/mariadb/documentation/storage-engines/spider/>
- <https://mariadb.org>
- <http://spiderformysql.com/>
- ralf.gebhardt@mariadb.com



Vielen Dank!

"MySQL is a registered trademark of Oracle and/or its affiliates. Other names may be trademarks of their respective owners SkySQL is not affiliated with MySQL."