

# MySQL Ecosystem in 2018

What's new & exciting around “World's most popular open source database”

---

Laurynas Biveinis

Percona Server for MySQL tech lead  
Big Data Conference Vilnius 2018  
2018-11-28



# Hi, I'm Laurynas

---

## Database C++ engineer at Percona & Percona Server for MySQL tech lead

- Been doing this since 2011
- Before that tried to do PhD in databases, failed

## Percona: “Unbiased Open Source Database Experts”

- Find and use the **right** tool for the job (even if it's not Percona Server for MySQL, *to my great disappointment*)
- Hence, in a good position to evaluate technologies

# MySQL: has been around since 1995

---

- Past the peak public hype
- Runs some of the world's largest OLTP databases
- Colorful yet healthy history of forks, acquisitions, politics
- Development is gaining momentum



# MySQL: popular in 2018

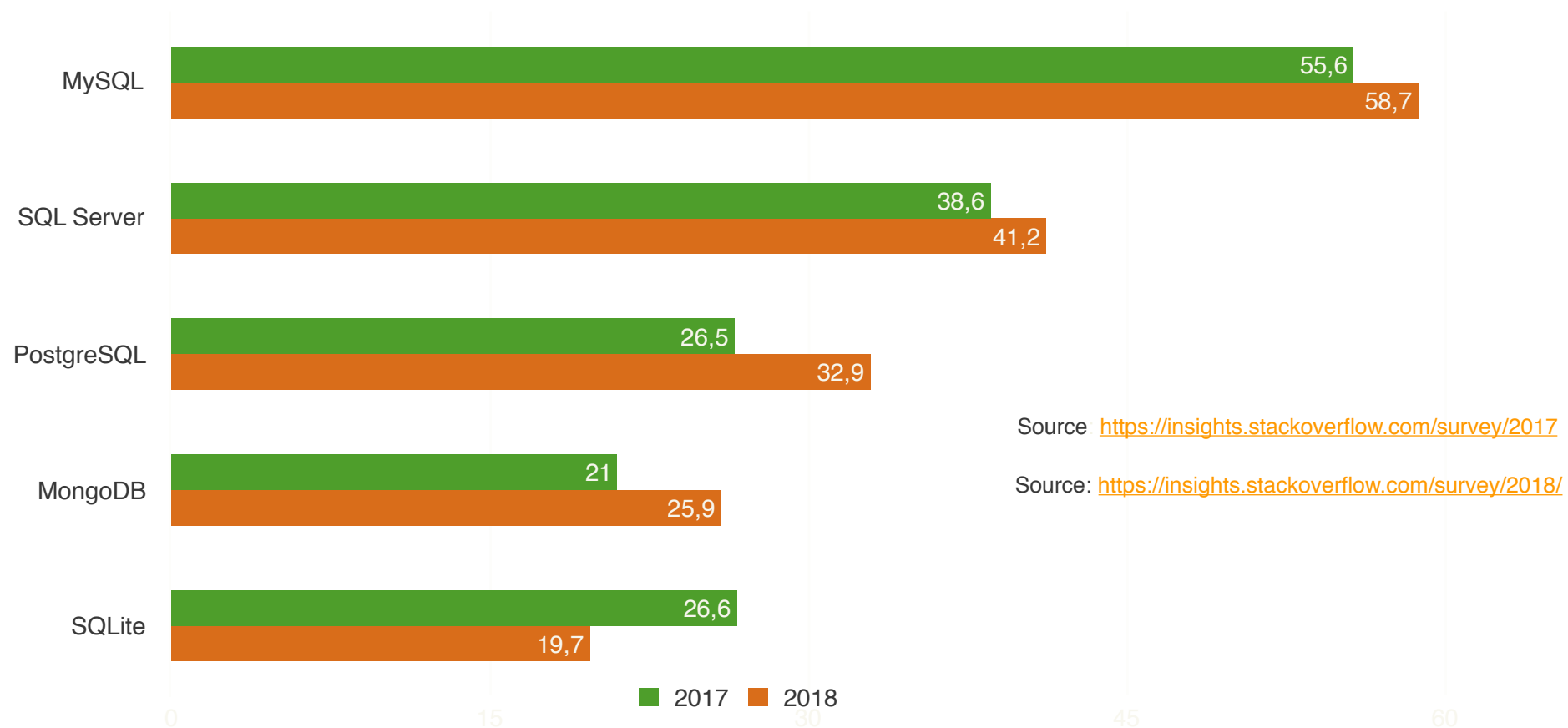
348 systems in ranking, November 2018

Rank			DBMS	Database Model	Score		
Nov 2018	Oct 2018	Nov 2017			Nov 2018	Oct 2018	Nov 2017
1.	1.	1.	Oracle +	Relational DBMS	1301.11	-18.16	-58.94
2.	2.	2.	MySQL +	Relational DBMS	1159.89	-18.22	-162.14
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1051.55	-6.78	-163.53
4.	4.	4.	PostgreSQL +	Relational DBMS	440.24	+20.85	+60.33
5.	5.	5.	MongoDB +	Document store	369.48	+6.30	+39.01

<https://db-engines.com/en/ranking>

# MySQL: popular in 2018

Stack Overflow Popularity



# 2018: MySQL 8.0 release

---

- Following MySQL 5.7 release in 2015 (the 1st one w/ doc store features)
- New data dictionary
- NOWAIT / SKIP LOCKED
- Instant ADD COLUMN
- partial in-place JSON updates, including replication
- JSON\_TABLE
- Common table expressions
- Window functions
- SET PERSIST
- ... too much to list here, go to <http://bit.ly/2TOKAQm>

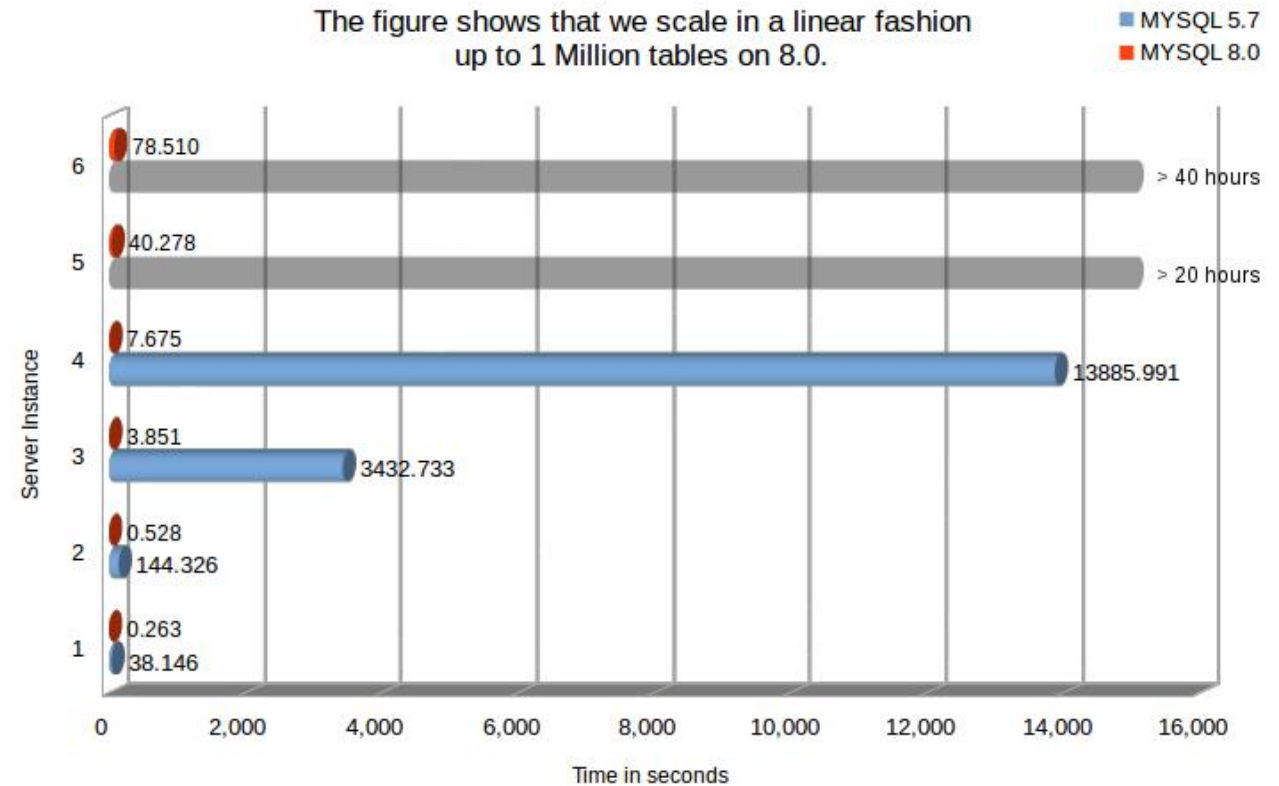
# MySQL 8.0: new data dictionary

---

- Remember FRM files? They are gone!
- Replaced by a transactional data dictionary: a set of internal InnoDB tables
- Finally crash-safe, atomic DDL statements
- No more metadata files means fewer file system bottlenecks
- One billion tables possible: <http://bit.ly/2DCXfjl>
- Fast INFORMATION\_SCHEMA queries
- However, if something goes wrong, your ls-cd shell skills will get you nowhere
  - ibd2sdi - human readable metadata

# MySQL 8.0 fast INFORMATION\_SCHEMA queries

- <http://bit.ly/2R98hkz>
- No longer filesystem scans
- Views over InnoDB tables in DD





# MySQL 8.0 hot row handling: NOWAIT & SKIP LOCKED

---

- Implemented in AliSQL (Alibaba patch) and Facebook patch
- Now in MySQL & MariaDB
- Return error immediately instead of row lock wait timeout:
  - `SELECT ... FOR UPDATE NOWAIT`
- Non-deterministically skip locked rows:
  - `SELECT ... FOR UPDATE SKIP LOCKED`
- Example use cases: online shopping inventory availability, cinema seat picker

# MySQL 8.0 instant add column

---

```
ALTER TABLE t1 ADD COLUMN d INT DEFAULT 1000,  
ALGORITHM=INSTANT;  
Query OK, 0 rows affected (0.07 sec)
```

# MySQL 8.0 partial JSON update

---

- JSON\_REPLACE on 1K doc, 10 byte update:
- in 5.7: DELETE + INSERT
- ... replication of JSON\_REPLACE in ROW: DELETE + INSERT!
- In 8.0: JSON\_REPLACE (JSON\_SET, JSON\_REMOVE) will do the delta in-place
- If binlog\_row\_value\_options=PARTIAL\_JSON, delta will be replicated in ROW too

# MySQL 8.0: JSON\_TABLE

---

- Convert JSON to a relational table, by providing column name, type & path:
- `SELECT * FROM JSON_TABLE(  
 '[{"x":1, "y":6}, {"x":2, "y":5}, {"x":3, "y":4}]', "$[*]"  
 COLUMNS (x INT PATH "$.x", y INT PATH "$.y")) AS j;`

```
+-----+  
| x | y |  
+---+---+  
| 1 | 6 |  
| 2 | 5 |  
| 3 | 4 |  
+---+---+
```

# MySQL 8.0: Common Table Expressions

---

```
WITH RECURSIVE cte (n) AS
(
  SELECT 1
  UNION ALL
  SELECT n + 1 FROM cte WHERE n < 5
)
SELECT * FROM cte;
```

n
1
2
3
4
5

# MySQL 8.0: Common Table Expressions

```
WITH RECURSIVE fibonacci (n, fib_n,  
next_fib_n) AS  
(  
  SELECT 1, 0, 1  
  UNION ALL  
  SELECT n + 1, next_fib_n, fib_n + next_fib_n  
    FROM fibonacci WHERE n < 10  
)  
SELECT * FROM fibonacci;
```

n	fib_n	next_fib_n
1	0	1
2	1	1
3	1	2
4	2	3
5	3	5
6	5	8
7	8	13
8	13	21
9	21	34
10	34	55

# MySQL 8.0: Window Functions

---

```
SELECT MONTH(date), SUM(sale),  
       AVG(SUM(sale)) OVER (ORDER BY MONTH(date)  
                           RANGE BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS sliding_avg  
FROM sales GROUP BY MONTH(date);
```

# MySQL 8.0: SET PERSIST

---

- In earlier versions, applying a config change:
  - SET GLOBAL variable = new\_value;
  - `$ vi my.cnf # I hope this is the right file and I didn't make a typo`
  - ... learn what broke on the next restart ...
- In 8.0
  - SET PERSIST variable = new\_value
- For a RO variable, to take effect on the next restart:
  - SET PERSIST\_ONLY variable = new\_value
  - RESTART (also a new SQL command in 8.0)
- No shell required



# MariaDB: the main MySQL-like alternative

---

- Concerns about Oracle ownership of MySQL
  - *which have not materialized*
- By now, a database on its own, no longer “a flavor of MySQL”
- Different (and diverging) feature set
- Not necessarily MySQL-compatible in replication setup



# 2018: MariaDB 10.3 is GA

---

- System-versioned tables
- Oracle PL/SQL compatibility
- Instant ADD COLUMN
- SEQUENCES
- ... lots more, <http://bit.ly/2E1zdzN>

# MariaDB 10.3: system-versioned tables

---

- CREATE TABLE ... WITH SYSTEM VERSIONING
- INSERT a row, then UPDATE, UPDATE, UPDATE it, and then:
- SELECT \*, ROW\_START, ROW\_END FROM table FOR SYSTEM\_TIME ALL;
  - a    b            ROW\_START            ROW\_END
  - 1    0            2018-11-28 10:48:04.435534            2018-11-28 10:48:04.437197
  - 1    1            2018-11-28 10:48:04.437197            2018-11-28 10:48:04.437505
  - 1    2            2018-11-28 10:48:04.437505            2038-01-19 05:14:07.999999
- SELECT \* FROM table FOR SYSTEM\_TIME AS OF TIMESTAMP '2018-10-01 00:00:00'
- Transactionally-consistent *and inconsistent*
- Space management provided too: PARTITION p HISTORY / DELETE HISTORY

# MariaDB 10.3: Oracle PL/SQL

---

- `--sql-mode=oracle`
- Changes stored procedure / stored function / cursor / LOOP / variable syntax, some function behavior, etc etc
- Adds PACKAGE support
- Empty string "" and NULL become identical
- Etc
- Might make migration easier?

# MariaDB 10.3: SEQUENCE

---

- CREATE SEQUENCE seq START WITH 1 INCREMENT BY 1;
- INSERT INTO TABLE (id, ...) VALUES (NEXT VALUE FOR seq, ...)
- At column definition:
  - NOT NULL DEFAULT NEXT VALUE FOR seq
  - AUTO\_INCREMENT-like but more flexible
- SELECT PREVIOUS VALUE for seq;

# Percona Server: MySQL improved

---

Patch (not fork) MySQL to add:

- Enterprise features for free (threadpool, PAM auth)
- Instrumentation
- Performance/scalability
- Selected new features
- <http://bit.ly/2DOg4Au>



PERCONA  
SERVER

# Percona Server in 2018

---

- Coming at the very end of 2018: Percona Server 8.0 GA, now at RC
- Data at Rest Encryption
- MyRocks: a new write- and space- optimized storage engine as GA



PERCONA  
SERVER

# Percona Server: Data at Rest Encryption

---

- MySQL 5.7: encryption of InnoDB tables, keys stored in a plaintext file
- MySQL 8.0: adds redo- & undo- log encryption
- MariaDB: also binlog encryption, various temp files, automatic key rotation, key storage in AWS
- Percona Server 5.7: similar to MariaDB but compatible with MySQL & key storage in Vault (not AWS)
- Both MariaDB and Percona Server can reasonably achieve *everything-is-encrypted* state
- <http://bit.ly/2zuRtxP>

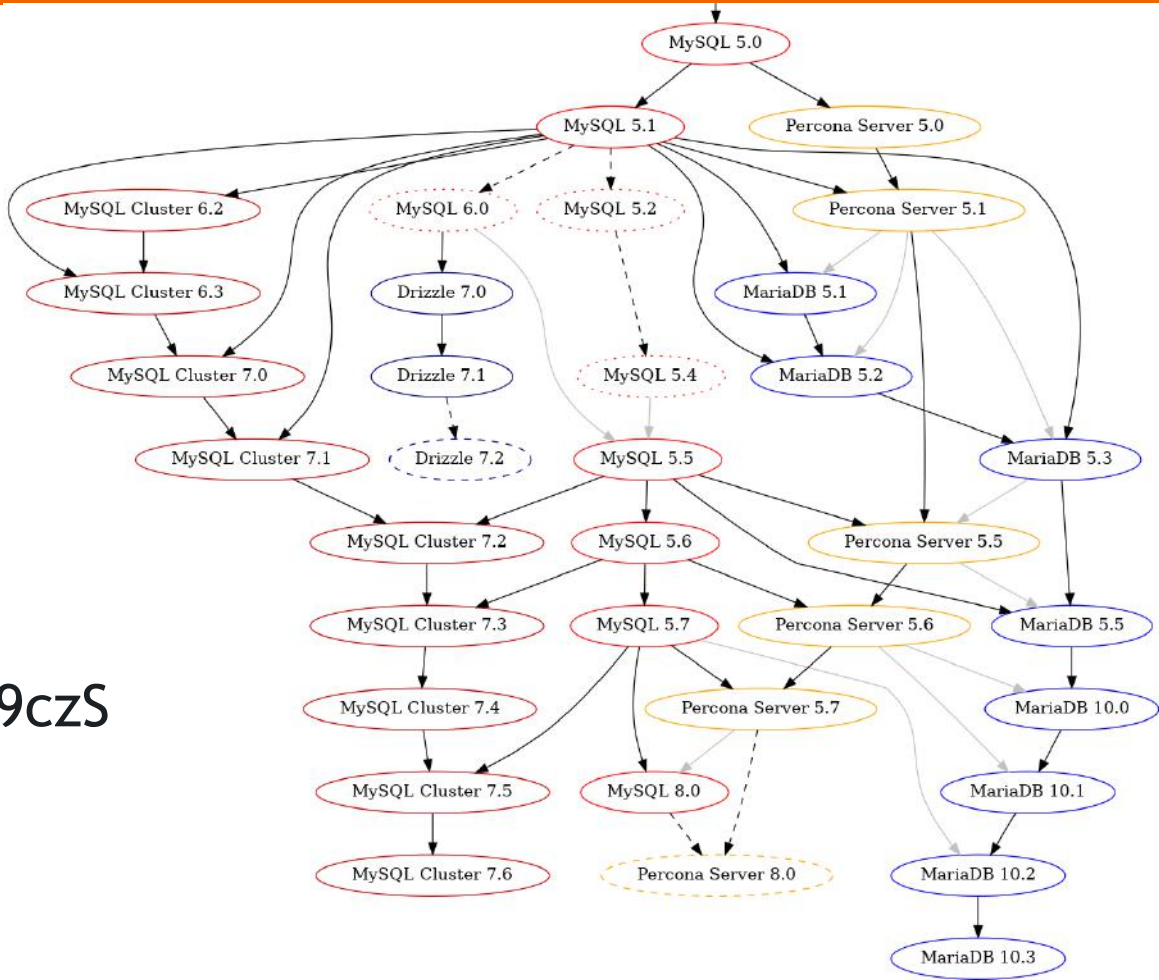


# Percona Server: MyRocks

---

- MyRocks: Facebook-made MySQL storage engine on the top of RocksDB
- RocksDB: Facebook-made key-value store library
  - Fork of Google LevelDB
- InnoDB: B-tree
  - Optimized for reads
- RocksDB: LSM-tree
  - Optimized for writes and disk space while reads stay OK-ish
- You probably don't want to run MySQL Facebook Patch directly yourself (even if you can)
- Percona (and MariaDB) integrated MyRocks for the general community

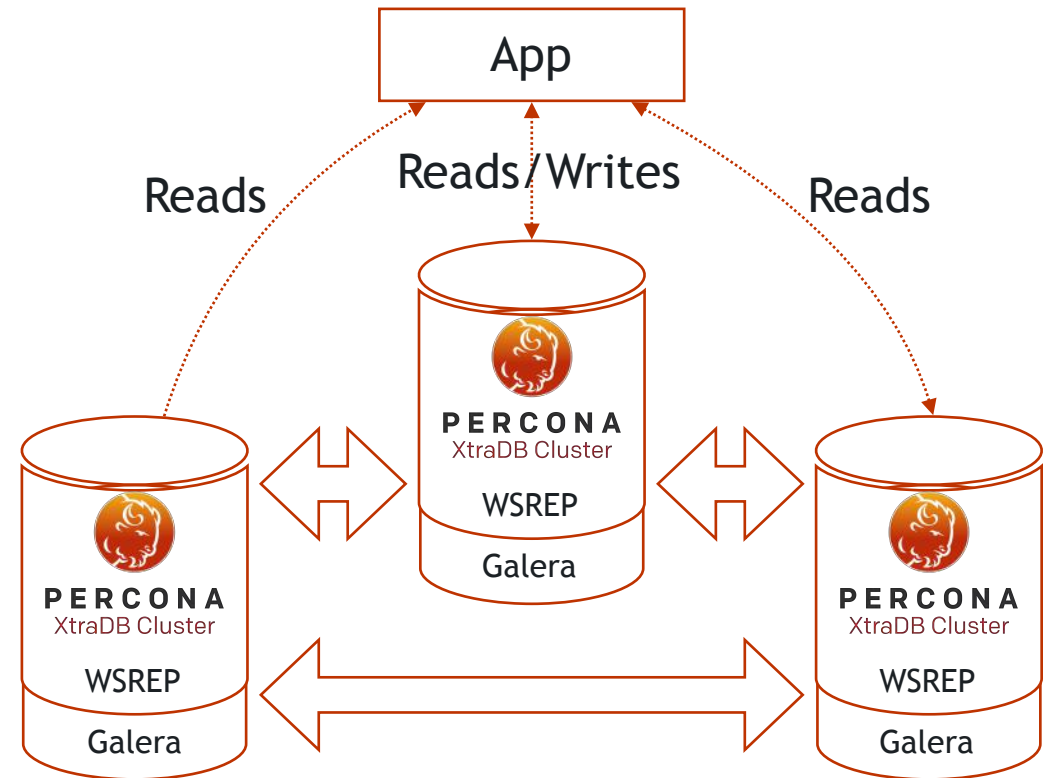
# What's the deal with all those forks?



<http://bit.ly/2Qq9czS>

# MySQL in 2018: clusters

- The built-in asynchronous replication does not make a cluster
  - HA? Multi-master?
- Existing (and actively-developed) clusters: Percona XtraDB Cluster & MariaDB
  - Based on galera library + Write Set replication patch
- New player: Group Replication



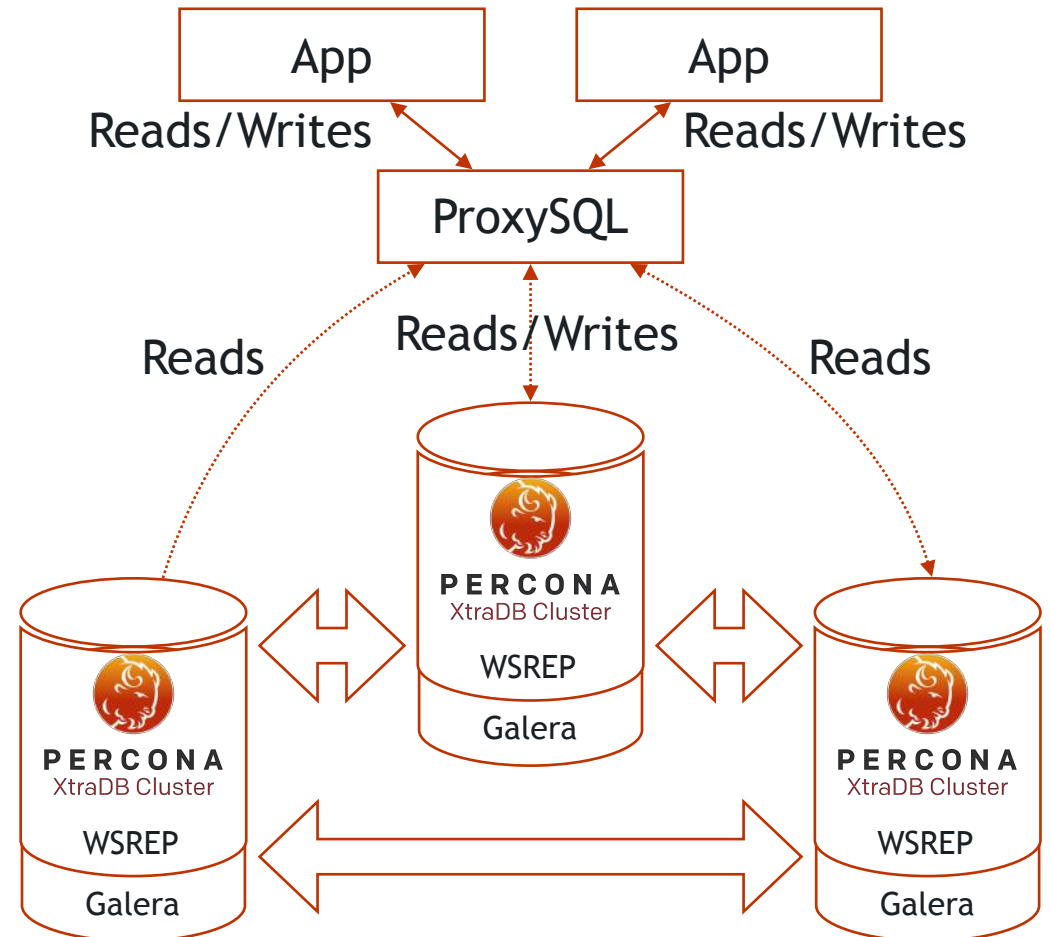
# Group Replication

---

- Oracle MySQL shared-everything cluster (not to be confused with shared-nothing MySQL Cluster)
- Went GA in 5.7 at the end of 2016
  - With number of issues that we couldn't agree with the GA qualification
- However, Oracle has been hard at work to fix them
- Lots of progress in 8.0
- Still, elephant in the room: no automated node provisioning
- If not GA, then getting there quickly

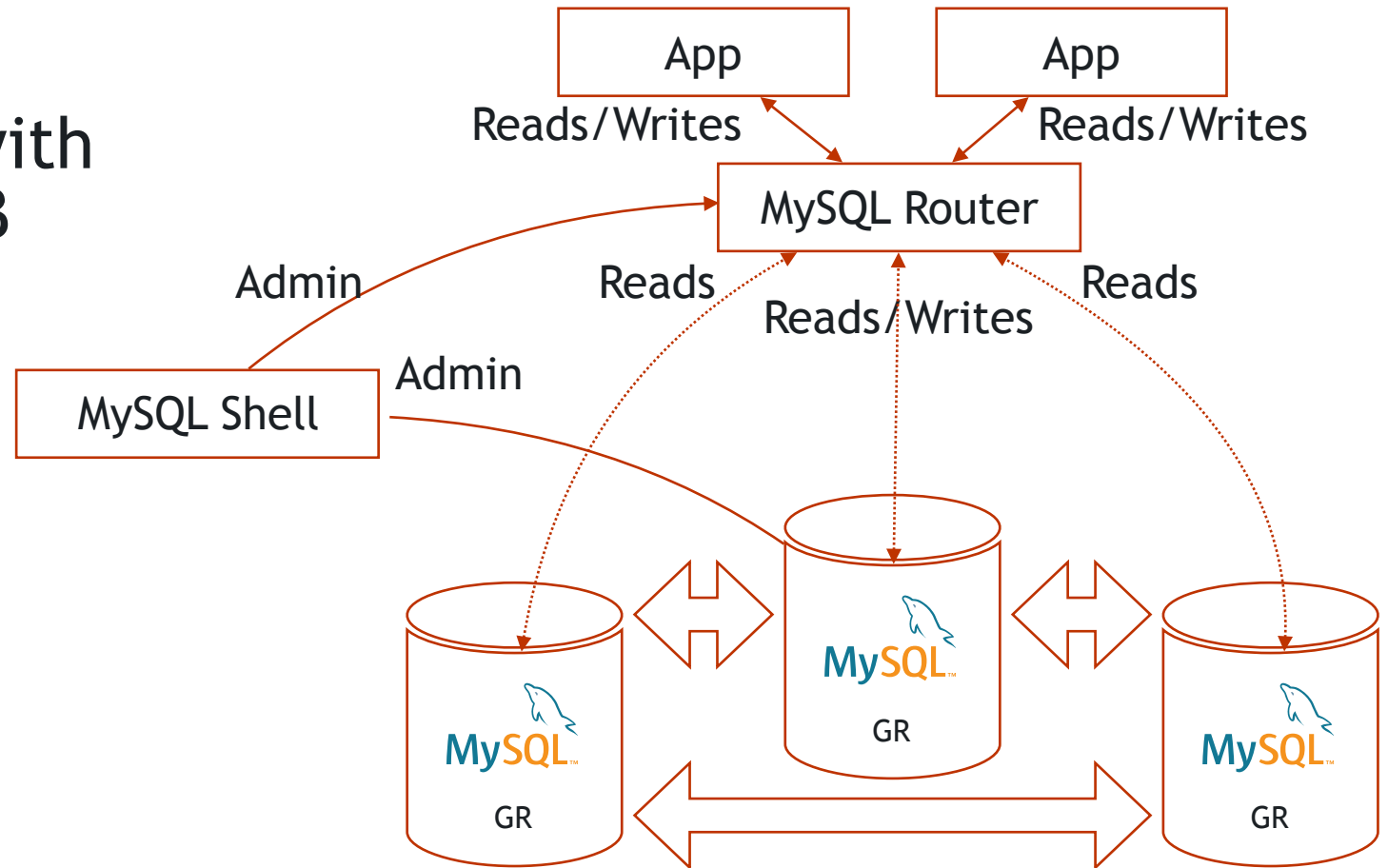
# What sits in the front of the cluster?

- Proxy does! If nothing else, to shield you from the “who is the primary now?” question
- “Best” option in 2018:
  - ProxySQL, 2.0 coming soon
- Other options:
  - MySQL Router
  - MariaDB MaxScale (mind the BS licence)



# (A side note on MySQL vision here)

- InnoDB Cluster
  - (not to be confused with MySQL Cluster, XtraDB Cluster, ... Cluster)
- Group Replication
- MySQL Router
- MySQL Shell



# MySQL in 2018: last but not least

---

- Physical backups:
  - Percona XtraBackup is the leader, being updated for 8.0
  - mariabackup if using MariaDB incompatible features such as encryption
- Percona Toolkit, MySQL Utilities
- MariaDB ColumnStore, mydumper, MHA, vitess, PRM on Pacemaker, MySQL Sandbox, DBaaS, Aurora, ...
- Trying to dockerize MySQL
- And then everyone is trying to figure out how to use Kubernetes
- ...

# MySQL in 2018: Thank you!

---

- join at [slido.com](https://www.slido.com) with #bigdata2018 for questions!