

# THE VALUE OF OPEN SOURCE

---

**MICHAEL "MONTY" WIDENIUS**

Entrepreneur, MariaDB Hacker, MariaDB CTO

MariaDB Corporation AB

**2019-09-25 Seoul**



# 11 Reasons Open Source is Better than Closed Source

- Using **open standards** (no lock in into proprietary standards)
- Resource friendly; OSS software tend to work on old hardware
- **Lower cost**; Usually 1/10 of closed source software
- No cost for testing the full software
- **Better documentation** and more troubleshooting resources
- **Better support**, in many cases directly from the developers
- Better security, auditability (no trap doors and more eye balls)
- **Better quality**; Developed together with users
- Better customizability; You can also participate in development
- **No vendor lock in**; More than one vendor can give support
- When using open source, you take **charge of your own future**

Note that using open source does **not** mean that you have to become a software producer!

# OPEN SOURCE, THE GOOD AND THE BAD

- Open source is a better way to develop software
  - More developers
  - More spread
  - Better code (in many cases)
  - Works good for projects that can freely used by a lot of companies in their production or products.
- It's very hard to create a profitable company developing an open source project.
  - Not enough money to pay developers.
  - Hard to get money and investors for most projects (except for infrastructure projects like libraries or daemon services).

# OPEN SOURCE IS NATURAL OR WHY OPEN SOURCE WORKS

- You use open source because it's less expensive (and re-usable)
- You solve your own problems and get free help and development efforts from others while doing it.
- You participate to increase your reputation (to get a new job?)
- You give patches back to not have to maintain them.

There are of course people that participate because they believe in the open source cause or want to help others, but these are a minority (and most still get paid for this).

# Debunking some open source FAD from thenextweb.com

- Open-source (OS) platforms can increase the risk of security breaches
- Confusing Complexity
- No vendor releasing updates
- Make sure the software is maintained and has a large community to carry on its support
- Open source typically doesn't have as great a set of training manuals and resources as the paid-and-packaged stuff
- Lack of customer support
- You run into a problem with figuring out which sources are making changes to the code you're using
- Certain closed-source programs not working well with open-source ones, your staff may not be familiar with the Open Source programs
- Open Source software, such as WordPress, can sometimes come with a steep learning curve
- You need to do outline a policy for your business or organization on your open-source usage to not get too many different components
- You need outline a policy for your business or organization on your OS usage to see the big picture.
- One of the appeals of open-source software is cost. However, many companies fail to calculate the time commitment necessary to run and maintain the open-source code

False   True   Confused (No difference with closed source)



## SUCCESS / STATE OF OPEN SOURCE

- Apple wouldn't exist without open source
- Most servers are running an open source operating system.
- Much of the new technology, like solutions for big data, are open source.

In 2007, before iPhone and Android, many of the open source users said:

“Open Source is winning. We own the data centers, there's only the applications left”

- Desktop is getting better, but most new applications for mobile are closed source:
- 2.7M apps in Google play, 2.2M apps for iPhone.
- F-Droid (open source software repository) has only 2.200 open source apps.

## REASONS FOR USERS TO TRUST A OPEN SOURCE VENDOR

- Open source vendors are more trustworthy as they depend on trust to survive
- **No vendor lock in.** Your investment in using the product is safe even if:
  - Vendor goes out of business
  - Vendor would surprisingly change business terms radically
  - Vendor would stop supporting the version of the product you are using
- If this would happen to a popular product, someone would go away with the code and start maintain it themselves
- Little risk for **hidden trap doors** as one can examine the product code



## BENEFITS FOR DEVELOPERS USING OPEN SOURCE VENDOR

- Easy to **get access**, look at and use the code
- Freedom to examine and **change** any part of the code to satisfy your business reasons, fix bugs or port to other systems.
- Freedom to **find anyone** to do the above
- Freedom to **use** (read, build and change) the code and redistribute the code in an open source environment.

# THE MAIN FREE / OPEN SOURCE LICENSES (VERY SIMPLE VIEW)

- **Public domain**
  - Gives user freedom to do anything, including changing your copyright and claim they wrote it.
- **BSD/Apache**
  - Gives user freedom to full use, but needs to keep copyright in the source code.
- **LGPL**
  - Gives user freedom to use freely, but if they distribute it changed, they need to publish the changes under LGP
- **GPL**
  - Gives user freedom to use it freely, but if they distribute it, they need to publish changes and their code under GPL. Used by MySQL and MariaDB
- **AGPL** (Optional addition to GPL V3)
  - Free usage, but users needs to publish the code and the code connected to it even if not distributed (like web server). Was used by MongoDB

# BUSINESS MODELS TO USE WITH OPEN SOURCE

- **Open-Core Model** - have an open-source core and sell closed-source features on top of it (e.g. SugarCRM)
- **Dual Licensing Model**- one product/project that gets licensed with a viral, GPL-style license and a commercial closed-source license (e.g. MySQL)
  - Another option is “**Business source**” or Time Delayed Open Source
  - Commons clause is a similar concept (Source available, pay for commercial use)
- **Services Models** - where you get to download a productized version of an open-source project and pay a fee for the support you get on it for new features. You can normally also pay for training, features etc.
- **Subscriptions** (Usually a combination of support, extended product lifetime and guaranteed updates)
- **Donations**, crowd funding or advertisements.
- Having a **non profit foundation** fund part of the development

# OPEN CORE

- Probably most popular way nowadays for business trying to do open source first time.
- What Oracle is doing with MySQL, EnterpriseDB with PostgreSQL and what MongoDB changed to.
- Is **not an open source business model**, because it uses closed components and most of the benefits open source developers expect from the product is gone:
  - You can't change, modify, port or redistribute the code
  - You are locked to one vendor
- You may be able to create a small developer community around the product but mainly by people that doesn't need the closed source extensions.
- For community developers, **the “worst” possible offer** is open core or closed source that used under subscription and stops working when subscription runs out

# DONATIONS AND CROWD FUNDING

- Crowd funding can be a good way to start a project, but not as good for continues developing of a product. (Hard to get money later for new features)
- Donations can work for big projects, like Wikipedia, but is very hard for smaller projects
  - MariaDB Foundation has received less than 5000 euro during the last 5 years in public donations (Paypal etc.)
  - Most support has come to the MariaDB foundation trough membership fees.
- Advertisements only works for VERY popular web sites/products and can't sustain a large development organization

## DUAL LICENSING

- Used first by Ghostscript. MySQL was the second product to use it.
- Can only be used when you have full rights to all the code.
- Give out the same code under two licenses, for example GPL or AGPL and normal commercial closed source.
- Companies that can't use the GPL (because they don't want to give out their code) can buy the closed source version.
- GPL only works well for infrastructure, easily embeddable products, like libraries or databases.
- AGPL can work with projects that is used in SAAS environment, but only if the customers does closed source changes to the product.
  - (Example: iText)

# BUSINESS SOURCE LICENSE (DELAYED OPEN SOURCE)

- Not an Open Source license, but gives the users similar advantages as Open Source.
- **Source code is available** from the start. Anyone has the right to copy, modify & distribute but can't use it commercially under some conditions you define.
- After X years the code automatically converts to some Open Source/Free license. The date has to be explicitly set in all source code files to avoid misunderstandings
- **Better than Open Core** as this removes the “one vendor” problem and the code will eventually be free.
- **Investor friendly** (as there is a way to force some users to pay).

Wikipedia has a totally different definition of delayed open source, needs to get corrected.

## REASONS FOR USERS TO TRUST A OPEN SOURCE VENDOR (AGAIN)

- Benefits the Business source license gives it's users are almost same as for open source:
- **No vendor lock in**
  - Anyone can develop or support the product.
- **All source code is available**
  - Anyone can fix bug or add new features.

The “only” caveat is that if you are using the BSL version in **production**, you **have** to pay for a limited time



# BUSINESS SOURCE PROJECTS USING IT

- MaxScale proxy from MariaDB Corporation (More products coming)
  - <https://mariadb.com/bsl>
  - <https://mariadb.com/bsl-faq-adopting>
- **Landoop Apache Kafka tools**
- **Cockroach** database
- Some early adopters, who tried BSL (BSL is still a new license)
  - Baasbox used it early, but switched to a Software as a service (SAAS)
  - RapidMiner used it early, but switched to AGPL
  - RhodeCode used it early but switched to community and enterprise (closed source) editions.

## COMMONS CLAUSE

- Source available, one need license for commercial usage
- Major difference to BSL: No conversion to Open Source
- Author: Heather Meeker
- Used (first?) by Redis Labs (not Redis) August 2018
  - Redis Labs changed in Feb 2019 to the RSAL license which is less restrictive

For more information see:

<https://commonsclause.com/>

[https://www.theregister.co.uk/2018/08/23/redis\\_database\\_license\\_change/](https://www.theregister.co.uk/2018/08/23/redis_database_license_change/)

<https://www.zdnet.com/article/open-source-licensing-war-commons-clause/>

## OTHER “SOURCE AVAILABLE” LICENSES

Redis Labs RSAL license:

- RSAL grants equivalent rights to permissive open source licenses for the vast majority of users. With RSAL, developers can use the software, modify the source code, integrate it with an application, and use, distribute or sell the application. The only restriction is that the application cannot be a database, a caching engine, a stream processing engine, a search engine, an indexing engine or an ML/DL/AI serving engine.

Confluent Community License:

- Allows you to freely download, modify, and redistribute the code (very much like Apache 2.0 does), but it does not allow you to provide the software as a SaaS offering

# WHY TO SUPPORT OPEN SOURCE PROJECTS

If your company are dependent on an open source product you should support it!

Please question yourself about your open source products:

- What is the cost of changing to another product if the product you are using disappears / not developed anymore?
- What is the cost if the product stops working and you can't get proper help to fix it in 1-2 weeks?
- How important is it for you that the product continues to evolve?

# HOW TO SUPPORT AN OPEN SOURCE COMPANY

- Buy their subscription or support offering!
  - In most cases you also get help with common problems, not only bugs
- Donate money to the projects
  - MariaDB Corporation has donated to valgrind, Connect engine, Spider, SQLines etc.
- Become a paying member in their foundation
  - The MariaDB Foundation ensures that MariaDB server is always free and that pull requests from developers are processed.
  - Use the projects other services (SAAS, etc)

# WHAT MADE MYSQL SUCCESSFUL?

- We were using it (for data warehousing and web)
- Internet was new and everyone needed a web-optimized DB
- “Virtual company” made it easy to find good people
- New “free” license scheme (this was before Open Source)
  - Free for most, a few have to pay
  - Second program (Ghostscript was first) to use dual licensing, MySQL first to do it with GPL.
- Very easy to install and use (15 minute rule)
  - Released source and tested binaries for most platforms
- Friendly and helpful towards community
  - I personally wrote 30,000+ emails during the first 5 years to help people with using MySQL
- Waited with investments until product was “good enough”
  - Needed, stable and easy to use product with right price at the right time.

# INTRODUCING MARIA(DB)



# Why MariaDB was created

“Save the People, Save the Product”

- To keep the MySQL talent together
- To ensure that a free version of MySQL always exists
- To get one community developed and maintained branch
- Work with other MySQL forks/branches to share knowhow and code



# MariaDB is guaranteed to always be open source

**The MariaDB Foundation was created** to ensure that anyone can be a **contributor** to the MariaDB project **on equal terms!**

The MariaDB Foundation is the **owner** of the main MariaDB server repositories on github

The Foundation can never to be controlled by a single entity or person

The Foundation is **not** about the MariaDB trademark or to decide upon the MariaDB roadmap!

# MariaDB Foundation core members

- Booking.com
- MariaDB Corporation
- Alibaba
- Visma
- Development bank of Singapore (DBS)
- Tencent games
- Tencent cloud
- IBM
- Microsoft

# MariaDB corporation

- Owner of the MariaDB trademark, except for 3 parts which the MariaDB Foundation owns.
- Employs 240+ people, of which 60+ are engineers working on MariaDB and related software (MaxScale, ColumnStore, Clustrix...)
- Have the best MariaDB engineers & most of the MariaDB captains (people with write access to MariaDB source).
- Is the biggest driver of the MariaDB project.
- Sells support, subscriptions and tools around MariaDB server.
- Financially stable with recent investments from EIF, Alibaba and ServiceNow

# Open development

- Anyone can participate in the MariaDB server development on equal terms!
- All development plans are in the MariaDB Jira and the MariaDB Knowledgebase (KB)
- Anyone can get write access to the Knowledgebase or the code repository (if you are good enough)
- One source repository for all server features and all tests
- MySQL is open core and main repository doesn't have all features and lacks a lot of tests
- More secure as security bugs are fixed at once (not delayed to quarterly security releases)
- We use Zulip, a modern communication platform anyone can join and discuss with all the active MariaDB developers and other MariaDB users.
- There are a lot of developers of MariaDB in a lot of different companies

# Easy upgrades

- MariaDB does not remove features
- Upgrades should take only a few seconds
- Upgrading from MySQL to MariaDB is easier than upgrading between two MySQL versions (disregarding MySQL 8.0)
- No dump and restore is ever needed between releases
- Data on disk is forward compatible
- All old MySQL and MariaDB clients works with older and newer versions of MariaDB

# MariaDB & MySQL Compatibility

- User level (data, API, replication, configuration files..) compatible with MySQL
- Drop in replacement up to MySQL 5.7
- With MySQL 8.0+ one has to use mysqldump to move to MariaDB
- More plugins, more storage engines, more features, faster, better code quality.
- **LGPL** C, ODBC and Java connectors. More coming!
- All MySQL connectors should work with MariaDB (except X-protocol)

# Frequent releases

## (Release early, release often)

MariaDB 5.1 (Feb 2010)

MariaDB 5.2 (Nov 2010)

MariaDB 5.3 (Apr 2012)

MariaDB 5.5 (Apr 2013)

MariaDB 10.0 (Mar 2014)

MariaDB 10.1 (Oct 2015)

MariaDB 10.2 (Apr 2017)

MariaDB 10.3 (May 2018)

MariaDB 10.4 (June 2019)

MariaDB 10.5 (June 2020)

Making builds free

Community features

New optimizer

Merge MySQL 5.5

Parallel replication

Galera, Encryption

Advanced features

Oracle compatibility, Spider

Security, Faster ALTER, Galera 4

More Oracle compatibility

# MARIADB DEVELOPER COMMUNITY

- **185,789** MariaDB commits
- **320** MariaDB branches
- **859** MariaDB releases
- **191** MariaDB contributors
  - 40 Working at MariaDB Corporation & Foundation
- **6360** articles in the Knowledgebase

Source: Github, May 2019





# Innovation happens first at MariaDB

- Virtual columns
- Microseconds
- Group commit and Parallel replication
- Multi-source replication
- Multi Master (Galera / Group-replication)
- Atomic writes (For FusionIO and Shannon)
- Roles
- Encryption at rest (for InnoDB)
- Window functions
- Common table expressions (CTE)

# More storage engines

- MyISAM and Aria, for movable non transactional storage
- Direct read and write access to other databases and storage formats with the CONNECT engine:
  - Access to other databases through JDBC and ODBC
  - Direct access to files in a lot of different formats like JSON, XML, HTML, INI, Access, MongoDB
- ColumStore for OLAP queries with petabyte of data
- Spider for partitioned data in separate storage nodes
- MyRocks for highly compressed data
- Lots of others (Sequence, OQgraph, Sphinx, S3)
- Clustrix, a distributed OLTP engine that provides right scaling (for 10.5+)

# Features only in MariaDB

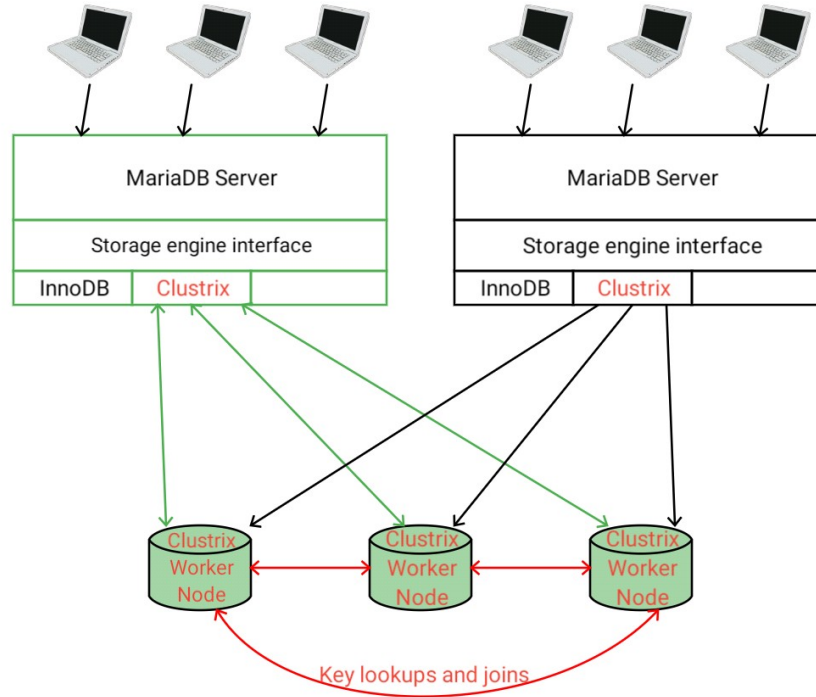
- Extended user statistics
- Segmented MyISAM key cache (faster multi user!)
- Thread pool (only in MySQL enterprise)
- More advanced optimizer
- Dynamic columns (Like JSON, but typesafe and more efficient storage)
- Working GTID (global transaction id) with domains
- DEFAULT with expressions (also for BLOB's)
- CHECK CONSTRAINT
- SEQUENCE's (CREATE SEQUENCE...)
- INTERSECT and EXCEPT

# Features only in MariaDB

- PL/SQL support (SET sql\_mode=ORACLE)
- Packages (Oracle feature)
- ROW data type
- TYPE OF
- Instant ADD COLUMN, DROP COLUMN etc for InnoDB
- Compressed columns and compressed binary log events
- Invisible columns
- Time machine (going back in time using binary log)
- System versioning (AS OF) for InnoDB

# What's coming in MariaDB 10.5+

## Clustrix, a storage engine with write scaling



**INNOVATIONS MADE  
POSSIBLE BY  
OPEN SOURCE**

# COMMUNITY COLLABORATION



# MariaDB is everywhere (Most distributions don't support MySQL anymore)

## Linux Distributions



## Cloud Services & Stacks



OPENSIFT

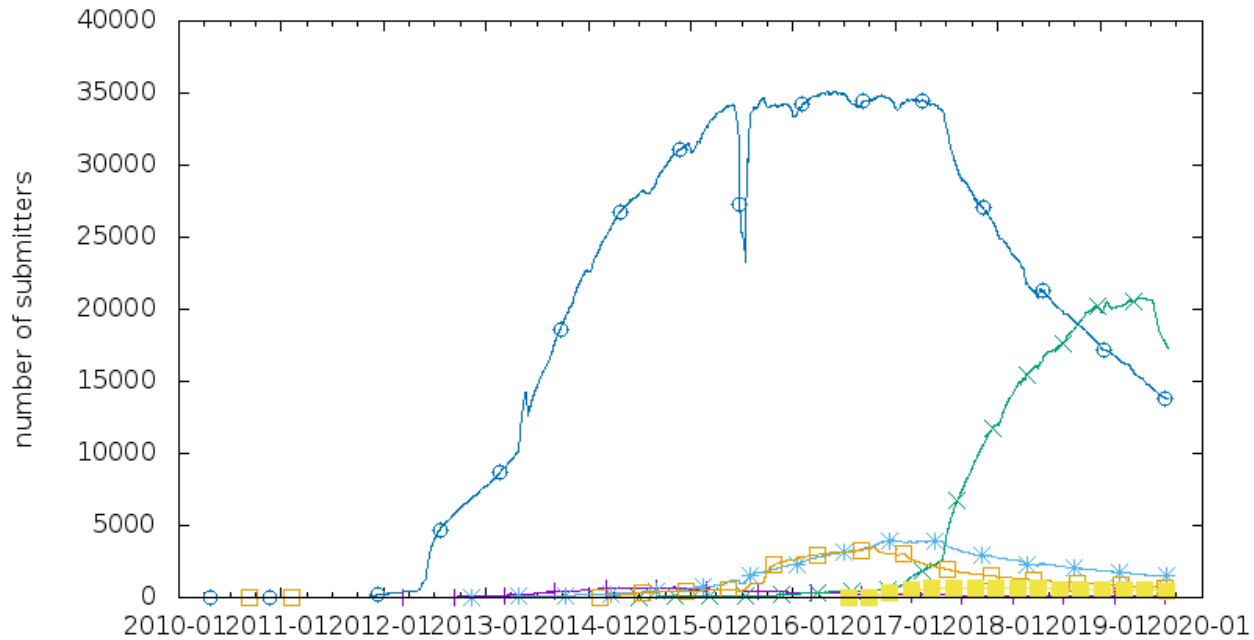


Pivotal Cloud Foundry®





popcon graph



date  
mariadb-server-5.5 installed +  
mariadb-server-10.1 installed x  
mariadb-server-10.0 installed \*  
mysql-server-5.6 installed □  
mysql-server-5.7 installed ■  
mysql-server-5.5 installed ○



## User's Choice Database of the Year

# LinuxQuestions.org

where Linux users come for help



# Reasons to switch to MariaDB today

- MariaDB is guaranteed to be always free!
- MariaDB is maintained by the people that originally created MySQL and has the best knowledge of the MySQL code.
- MariaDB is binary compatible (data and API) with MySQL up to 5.7, so its trivial to replace MySQL with MariaDB (minutes).
- **Faster queries** thanks to optimized InnoDB, ColumnStore, MyRocks, a much better optimizer
- **More features** like true parallel replication, better statistics, dynamic columns, encryption and many storage engines.
- Open source development: **Anyone can be part of the development** at all stages. Developer meetings are public!
- **Less risk**, as MariaDB will not remove features (like MySQL is doing)

# Summary: What made MySQL successful?

- We were using it (for data warehousing and web)
- Internet was new and everyone needed a web-optimized DB
- “Virtual company” made it easy to find good people
- New “free” license scheme (this was before Open Source)
- Free for most, a few have to pay
- Second program (ghostscript was first) to use dual licensing, MySQL first to do it with GPL.
- Very easy to install and use (15 minute rule)
- Released source and tested binaries for most platforms
- Friendly and helpful towards community
  - Monty wrote 30,000+ emails during the first 5 years to help people using MySQL
- Waited with investments until product was “good enough”
- MySQL was a needed, stable and easy to use product with the right price

# THANK YOU