Community vs. Enterprise: How not to piss off your community (and still be profitable)

Colin Charles, GrokOpen
colin@grokopen.com | @bytebot on Twitter

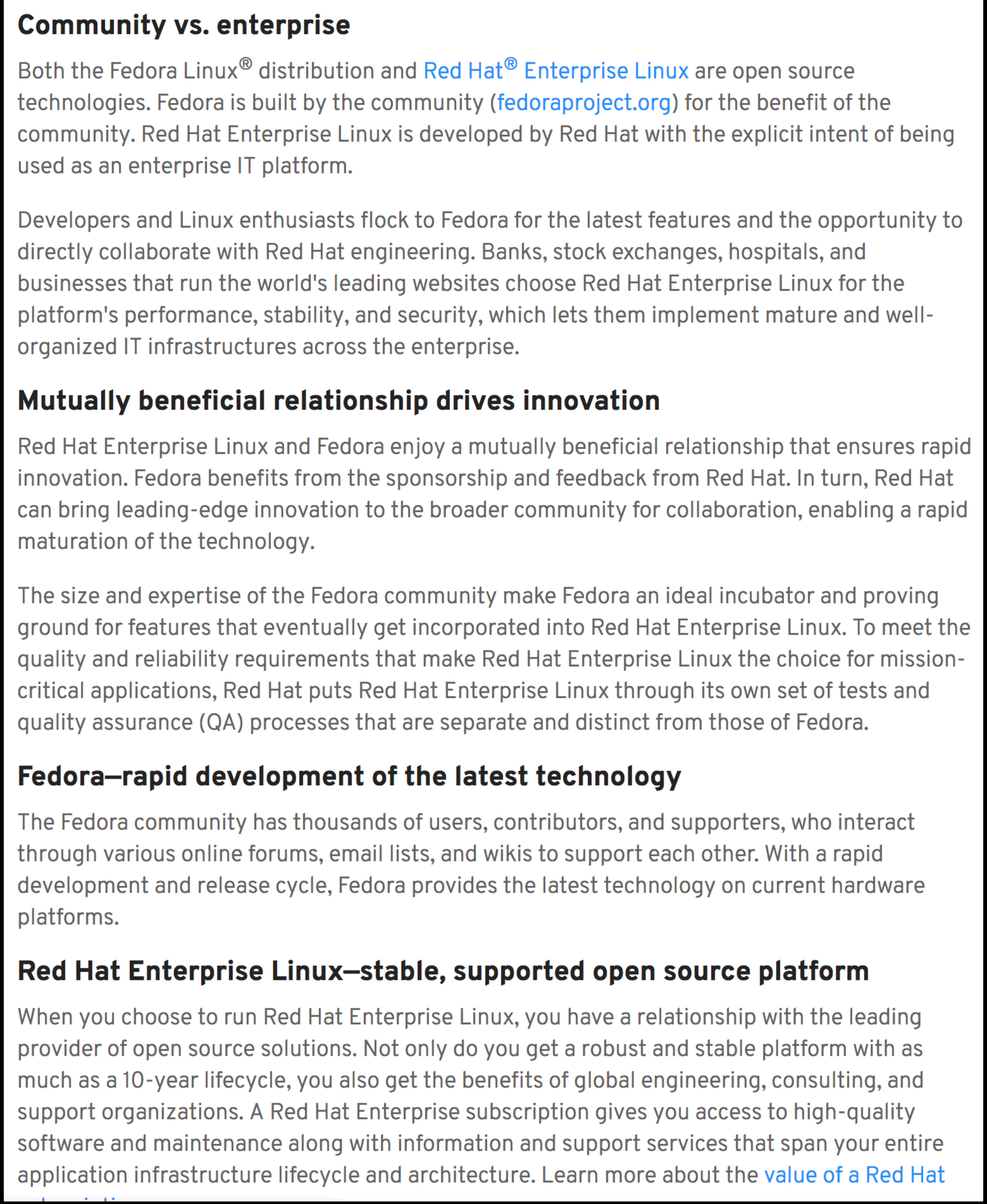
Open Source Strategy Forum, London, England, United Kingdom
14 November 2018

Premise

You have a **popular** open source project. You feel that its time to **make money** from it, so you start a **corporation** around it. You **raise venture capital**. Your open source project flourishes with development funded by money in the bank. You hire enough folk in sales to convince the upper echelons that it is time to make an "**enterprise spin-off**" that is better than the currently "stable" community edition. Your enterprise team goes all out to state that their version is better than the community version. The community gets annoyed but the code is significantly complex that forking might not be an option today. Rewind.

Red Hat: Fedora & Red Hat Enterprise Linux

- First… there was Red Hat Linux
- Then there was Fedora Core 1 and Red Hat Linux Beta 1 in July 2003
- Fedora Core 3 —> RHEL 4 (similarly, Fedora Core 6 —> RHEL 5, Fedora 12/13 —> RHEL 6, Fedora 19/20 —> RHEL 7)



Community vs. enterprise

Both the Fedora Linux® distribution and Red Hat® Enterprise Linux are open source technologies. Fedora is built by the community (fedoraproject.org) for the benefit of the community. Red Hat Enterprise Linux is developed by Red Hat with the explicit intent of being used as an enterprise IT platform.

Developers and Linux enthusiasts flock to Fedora for the latest features and the opportunity to directly collaborate with Red Hat engineering. Banks, stock exchanges, hospitals, and businesses that run the world's leading websites choose Red Hat Enterprise Linux for the platform's performance, stability, and security, which lets them implement mature and well-organized IT infrastructures across the enterprise.

Mutually beneficial relationship drives innovation

Red Hat Enterprise Linux and Fedora enjoy a mutually beneficial relationship that ensures rapid innovation. Fedora benefits from the sponsorship and feedback from Red Hat. In turn, Red Hat can bring leading-edge innovation to the broader community for collaboration, enabling a rapid maturation of the technology.

The size and expertise of the Fedora community make Fedora an ideal incubator and proving ground for features that eventually get incorporated into Red Hat Enterprise Linux. To meet the quality and reliability requirements that make Red Hat Enterprise Linux the choice for mission-critical applications, Red Hat puts Red Hat Enterprise Linux through its own set of tests and quality assurance (QA) processes that are separate and distinct from those of Fedora.

Fedora—rapid development of the latest technology

The Fedora community has thousands of users, contributors, and supporters, who interact through various online forums, email lists, and wikis to support each other. With a rapid development and release cycle, Fedora provides the latest technology on current hardware platforms.

Red Hat Enterprise Linux—stable, supported open source platform

When you choose to run Red Hat Enterprise Linux, you have a relationship with the leading provider of open source solutions. Not only do you get a robust and stable platform with as much as a 10-year lifecycle, you also get the benefits of global engineering, consulting, and support organizations. A Red Hat Enterprise subscription gives you access to high-quality software and maintenance along with information and support services that span your entire application infrastructure lifecycle and architecture. Learn more about the value of a Red Hat

What did Red Hat get right?

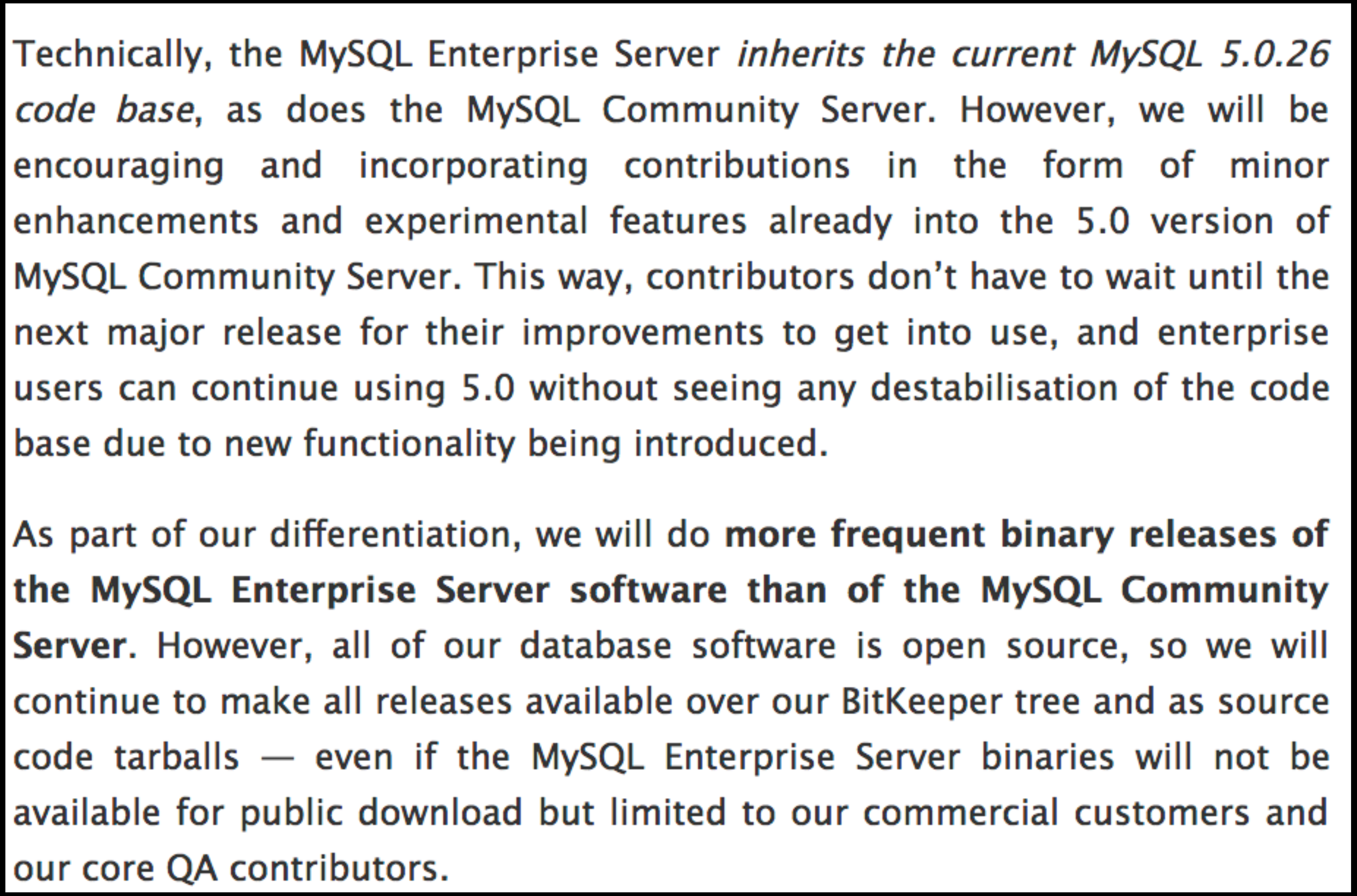
- Not naming it “Red Hat Linux Community” and “Red Hat Linux Enterprise”
    - Fedora and RHEL
- Less confusion
- Strong branding
- Helps the sales team out as well (as well as marketing)
- Community not seen as a *second class citizen* (with own branding, logos, events, etc.)

MySQL

- Became GPL in 2000
- Successfully sold OEM licenses as the connectors became GPL with the FOSS License Exception (previously LGPL) — https://www.mysql.com/about/legal/licensing/foss-exception/
    - now governed by the Universal FOSS Exception — https://oss.oracle.com/licenses/universal-foss-exception/
- MySQL Network Binaries, then a MySQL Community vs Enterprise split
- Dual-licensed — GPLv2 and commercial

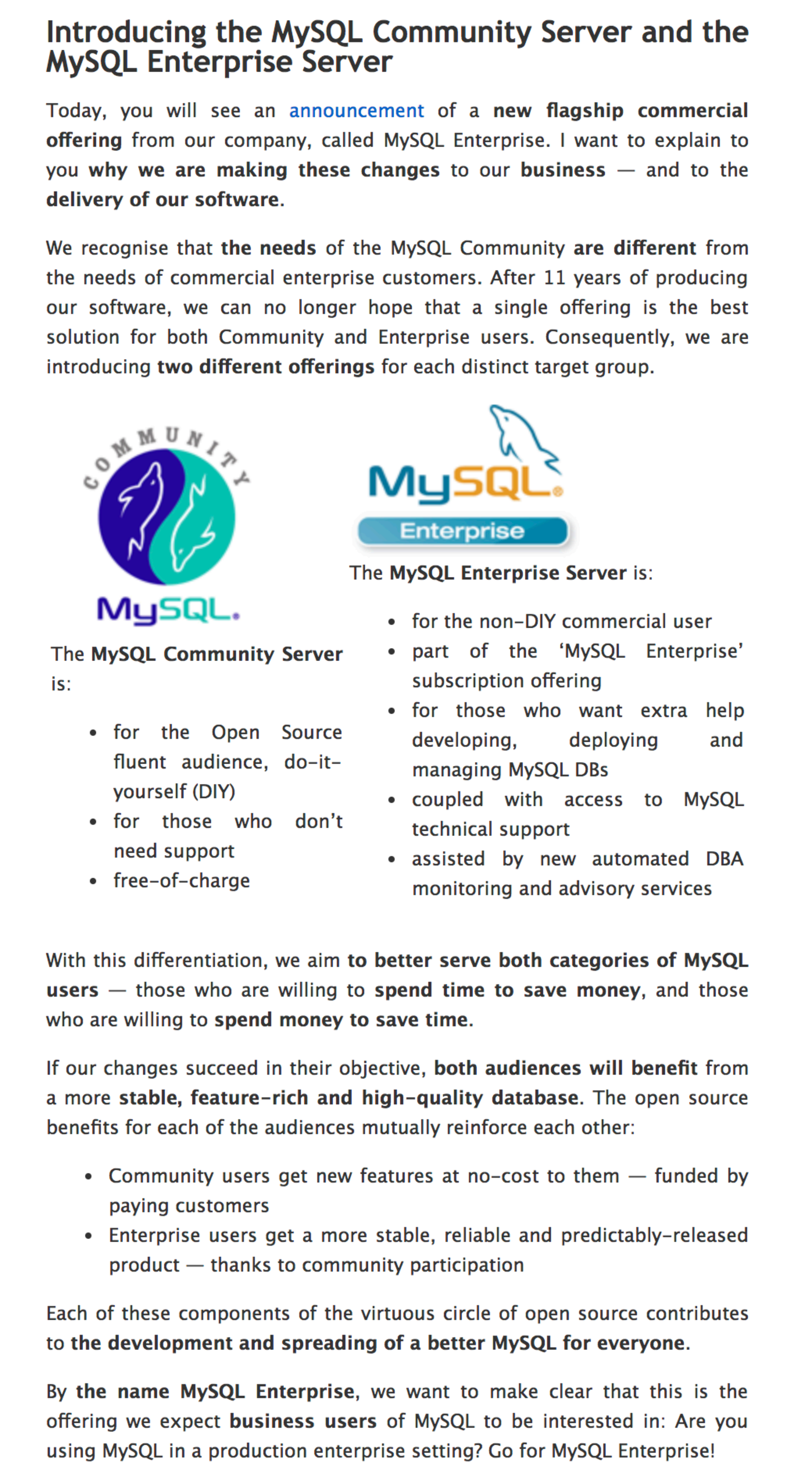
What did you get with MySQL Enterprise?

- Initially, nothing except a binary that said it was so!



Technically, the MySQL Enterprise Server *inherits the current MySQL 5.0.26 code base*, as does the MySQL Community Server. However, we will be encouraging and incorporating contributions in the form of minor enhancements and experimental features already into the 5.0 version of MySQL Community Server. This way, contributors don't have to wait until the next major release for their improvements to get into use, and enterprise users can continue using 5.0 without seeing any destabilisation of the code base due to new functionality being introduced.

As part of our differentiation, we will do **more frequent binary releases of the MySQL Enterprise Server software than of the MySQL Community Server**. However, all of our database software is open source, so we will continue to make all releases available over our BitKeeper tree and as source code tarballs — even if the MySQL Enterprise Server binaries will not be available for public download but limited to our commercial customers and our core QA contributors.



Introducing the MySQL Community Server and the MySQL Enterprise Server

Today, you will see an announcement of a **new flagship commercial offering** from our company, called MySQL Enterprise. I want to explain to you **why we are making these changes** to our **business** — and to the **delivery of our software**.

We recognise that **the needs** of the MySQL Community **are different** from the needs of commercial enterprise customers. After 11 years of producing our software, we can no longer hope that a single offering is the best solution for both Community and Enterprise users. Consequently, we are introducing **two different offerings** for each distinct target group.

The **MySQL Community Server** is:

- for the Open Source fluent audience, do-it-yourself (DIY)
- for those who don't need support
- free-of-charge

The **MySQL Enterprise Server** is:

- for the non-DIY commercial user
- part of the 'MySQL Enterprise' subscription offering
- for those who want extra help developing, deploying and managing MySQL DBs
- coupled with access to MySQL technical support
- assisted by new automated DBA monitoring and advisory services

With this differentiation, we aim **to better serve both categories of MySQL users** — those who are willing to **spend time to save money**, and those who are willing to **spend money to save time**.

If our changes succeed in their objective, **both audiences will benefit** from a more **stable, feature-rich and high-quality database**. The open source benefits for each of the audiences mutually reinforce each other:

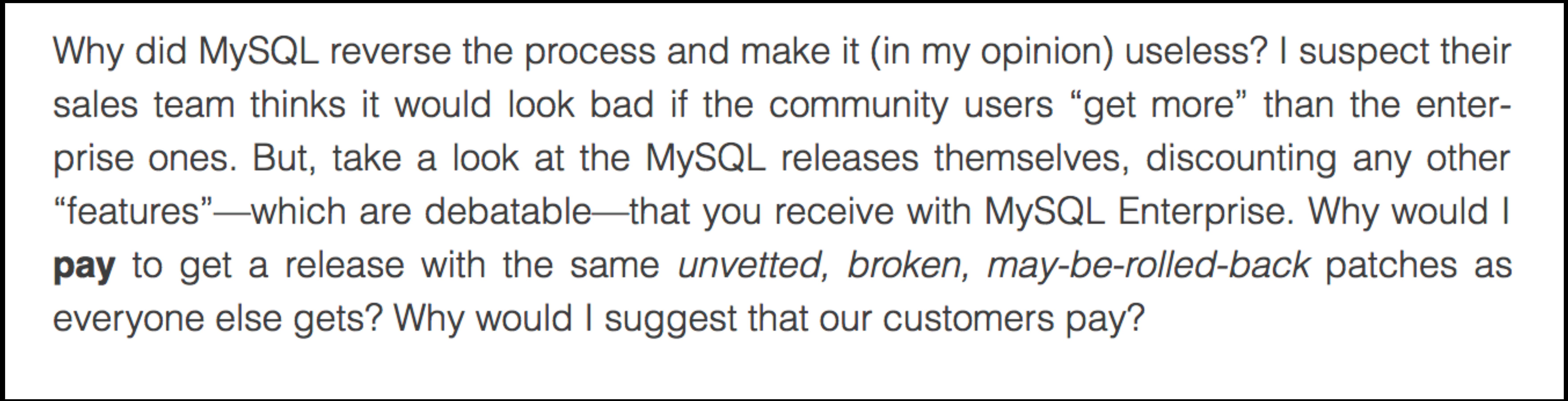
- Community users get new features at no-cost to them — funded by paying customers
- Enterprise users get a more stable, reliable and predictably-released product — thanks to community participation

Each of these components of the virtuous circle of open source contributes to **the development and spreading of a better MySQL for everyone**.

By **the name MySQL Enterprise**, we want to make clear that this is the offering we expect **business users** of MySQL to be interested in: Are you using MySQL in a production enterprise setting? Go for MySQL Enterprise!

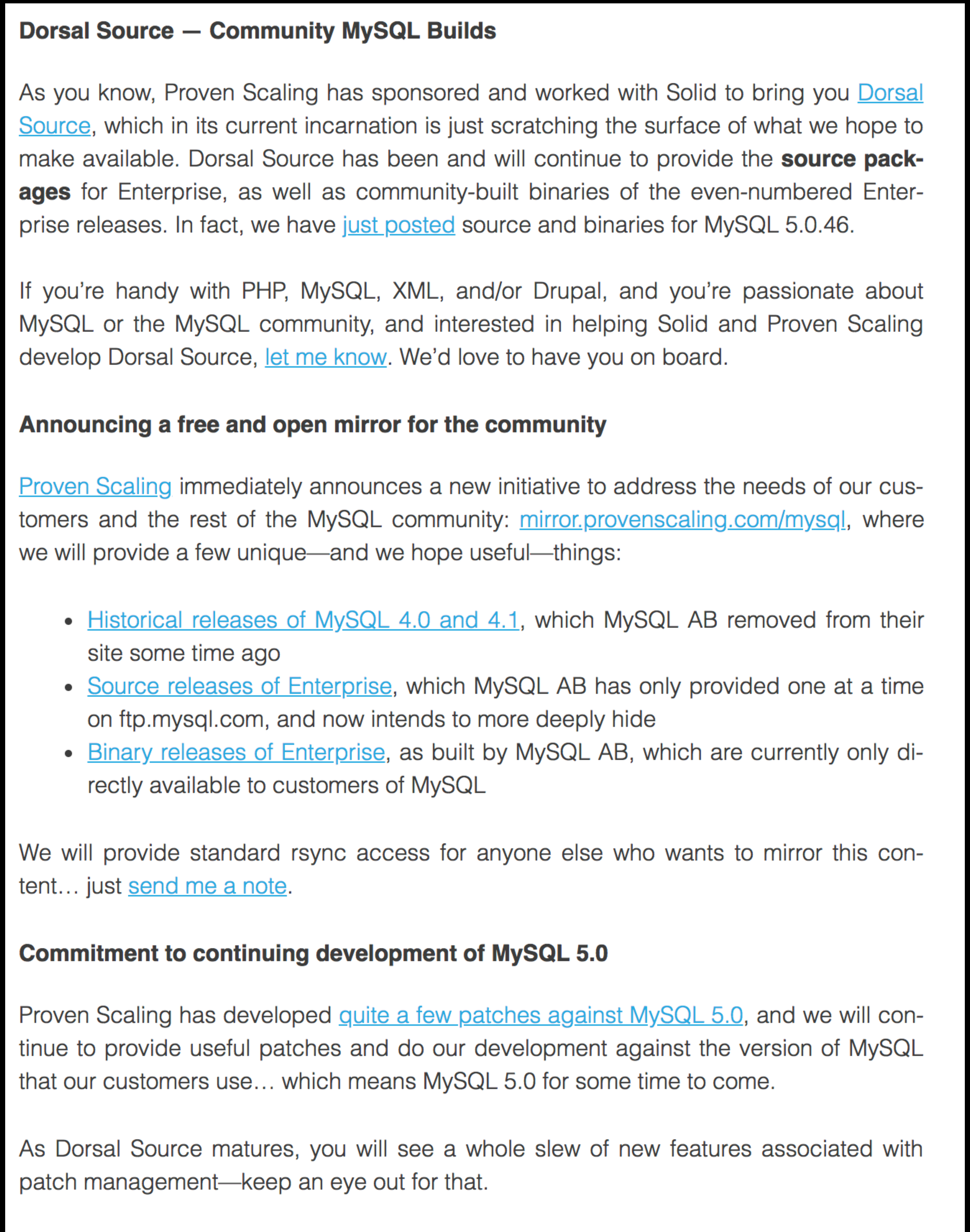
Community reacts

- Non-vetted code going into the Enterprise release, making it less stable (and require reversions!)
  - https://blog.jcole.us/2007/05/14/breakdown-in-mysql-enterprise-process/
  - Reverse “Red Hat/Fedora” model seems useless



Why did MySQL reverse the process and make it (in my opinion) useless? I suspect their sales team thinks it would look bad if the community users “get more” than the enterprise ones. But, take a look at the MySQL releases themselves, discounting any other “features”—which are debatable—that you receive with MySQL Enterprise. Why would I **pay** to get a release with the same *unvetted, broken, may-be-rolled-back* patches as everyone else gets? Why would I suggest that our customers pay?

Even better…



Dorsal Source — Community MySQL Builds

As you know, Proven Scaling has sponsored and worked with Solid to bring you Dorsal Source, which in its current incarnation is just scratching the surface of what we hope to make available. Dorsal Source has been and will continue to provide the **source packages** for Enterprise, as well as community-built binaries of the even-numbered Enterprise releases. In fact, we have just posted source and binaries for MySQL 5.0.46.

If you're handy with PHP, MySQL, XML, and/or Drupal, and you're passionate about MySQL or the MySQL community, and interested in helping Solid and Proven Scaling develop Dorsal Source, let me know. We'd love to have you on board.

Announcing a free and open mirror for the community

Proven Scaling immediately announces a new initiative to address the needs of our customers and the rest of the MySQL community: mirror.provenscaling.com/mysql, where we will provide a few unique—and we hope useful—things:

- Historical releases of MySQL 4.0 and 4.1, which MySQL AB removed from their site some time ago
- Source releases of Enterprise, which MySQL AB has only provided one at a time on ftp.mysql.com, and now intends to more deeply hide
- Binary releases of Enterprise, as built by MySQL AB, which are currently only directly available to customers of MySQL

We will provide standard rsync access for anyone else who wants to mirror this content… just send me a note.

Commitment to continuing development of MySQL 5.0

Proven Scaling has developed quite a few patches against MySQL 5.0, and we will continue to provide useful patches and do our development against the version of MySQL that our customers use… which means MySQL 5.0 for some time to come.

As Dorsal Source matures, you will see a whole slew of new features associated with patch management—keep an eye out for that.

Saved by acquisition!

- Sole differentiator: MySQL Enterprise Monitor (a product that was being built)
- When Oracle purchased Sun, could integrated MySQL Enterprise Backup (since Oracle had purchased InnoBase in 2005)
- Plugins: authentication (PAM, Active Directory), Audit plugin, threadpool, encryption plugins

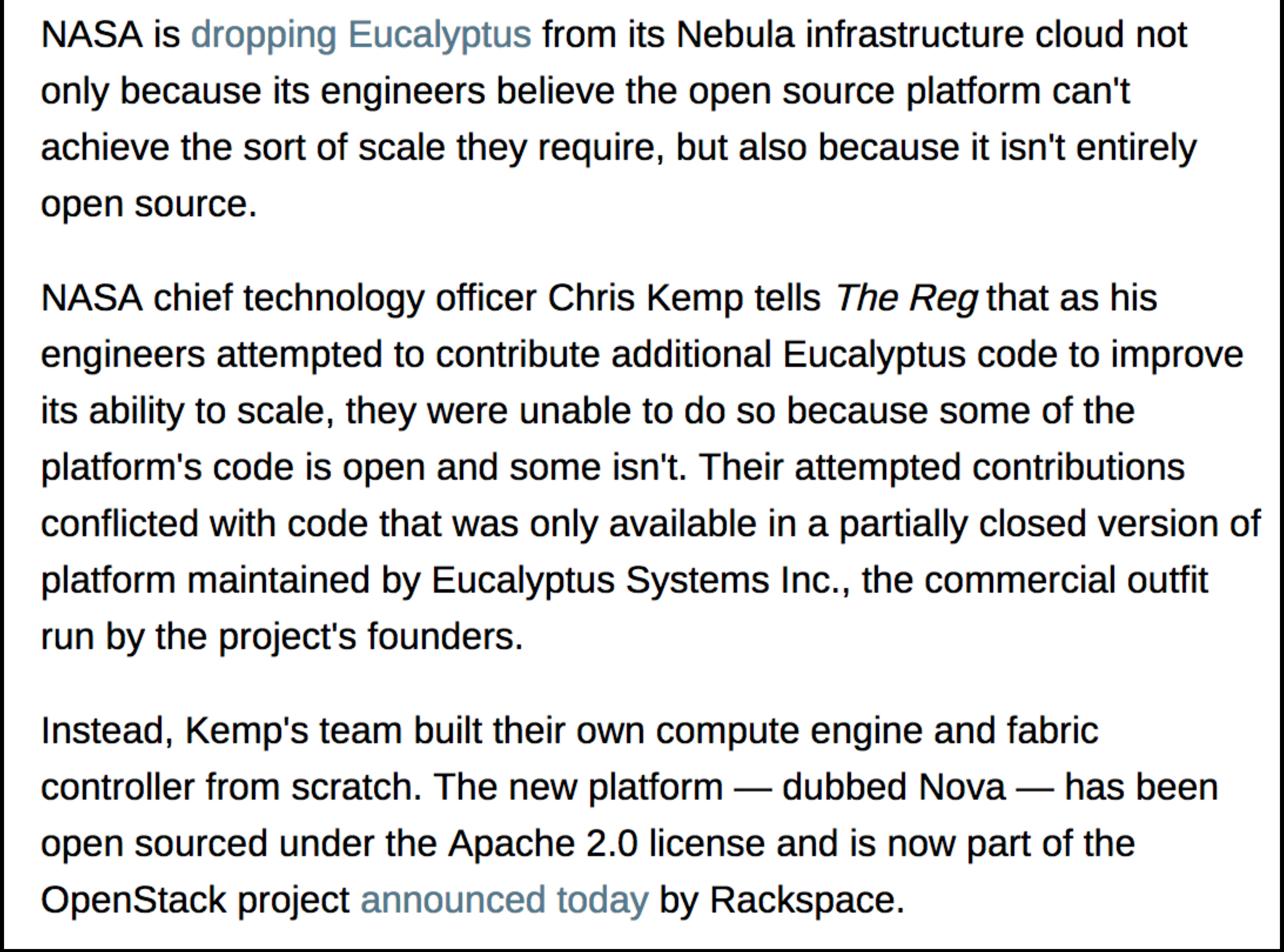
Where did MySQL go wrong?

- Never underestimate your community of users and developers
- Never underestimate the great QA the community was providing
- Don’t necessarily listen to the sales mantra that the community version is the untested worse variant of the enterprise variant
- No individual branding (because it was better to run on the MySQL name which was everywhere)
- Provide significant value when you do an Enterprise offering

Eucalyptus

- Was an open source compatible API alternative to Amazon Web Services baked into Ubuntu Karmic Koala as well
- Decided on an Open Core model by June 2010
    - "We deliver a fully functional cloud with Eucalyptus software. You can download it on a GPL v3 license. But, additionally, we provide enterprise features only if you pay for them ... it's open core”

Eucalyptus births OpenStack



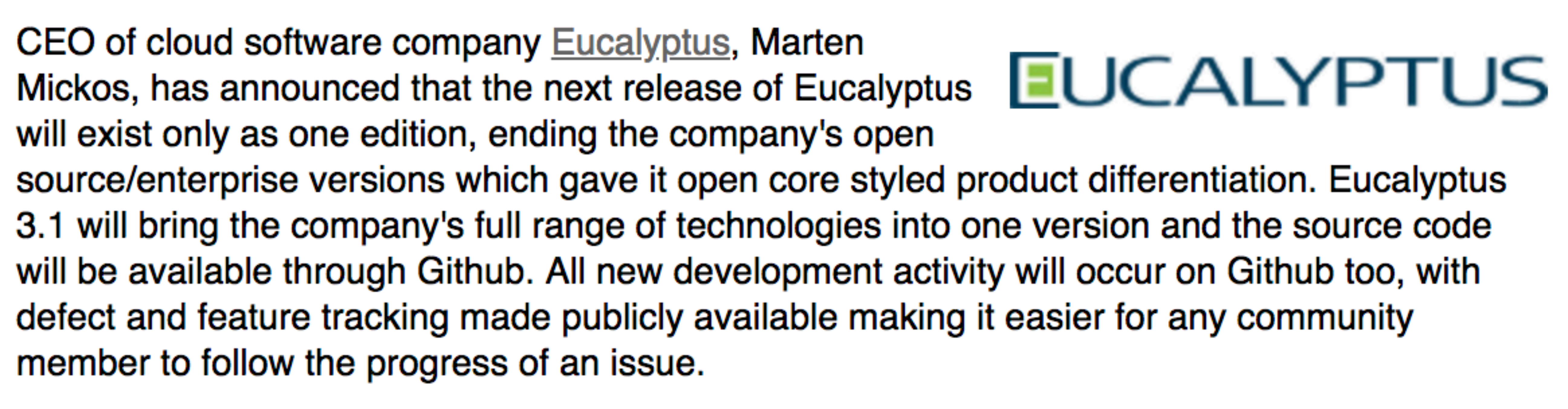
NASA is dropping Eucalyptus from its Nebula infrastructure cloud not only because its engineers believe the open source platform can't achieve the sort of scale they require, but also because it isn't entirely open source.

NASA chief technology officer Chris Kemp tells *The Reg* that as his engineers attempted to contribute additional Eucalyptus code to improve its ability to scale, they were unable to do so because some of the platform's code is open and some isn't. Their attempted contributions conflicted with code that was only available in a partially closed version of platform maintained by Eucalyptus Systems Inc., the commercial outfit run by the project's founders.

Instead, Kemp's team built their own compute engine and fabric controller from scratch. The new platform — dubbed Nova — has been open sourced under the Apache 2.0 license and is now part of the OpenStack project announced today by Rackspace.

Eucalyptus goes open again

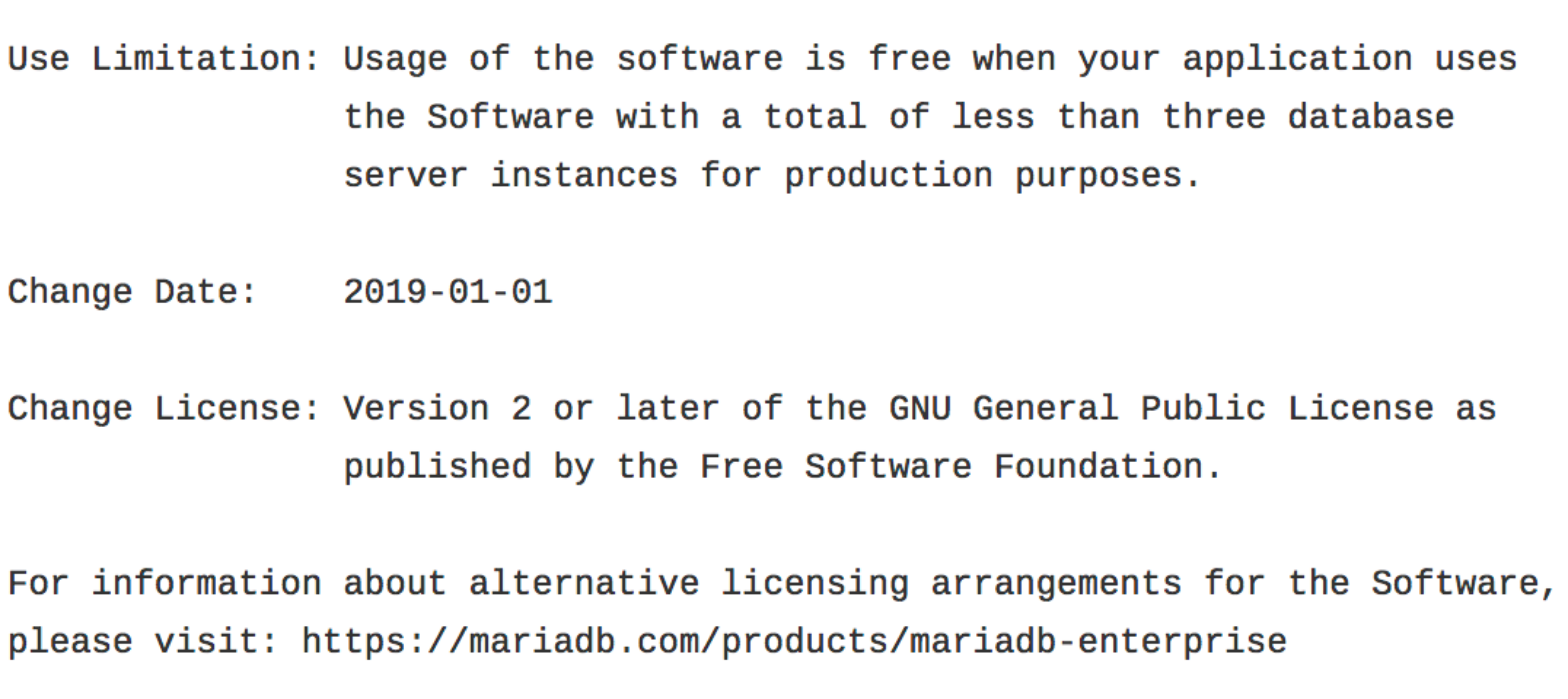
- June 2012: back to full open source
- September 2014: sold to HP



CEO of cloud software company Eucalyptus, Marten Mickos, has announced that the next release of Eucalyptus will exist only as one edition, ending the company's open source/enterprise versions which gave it open core styled product differentiation. Eucalyptus 3.1 will bring the company's full range of technologies into one version and the source code will be available through Github. All new development activity will occur on Github too, with defect and feature tracking made publicly available making it easier for any community member to follow the progress of an issue.

MariaDB MaxScale

- The market for proxies in the MySQL space:
  - Oracle MySQL provides MySQL Router (GPLv2)
  - ProxySQL (GPLv3)
  - MariaDB Server MaxScale (GPLv2 then changed to the Business Source License overnight in the same repository)

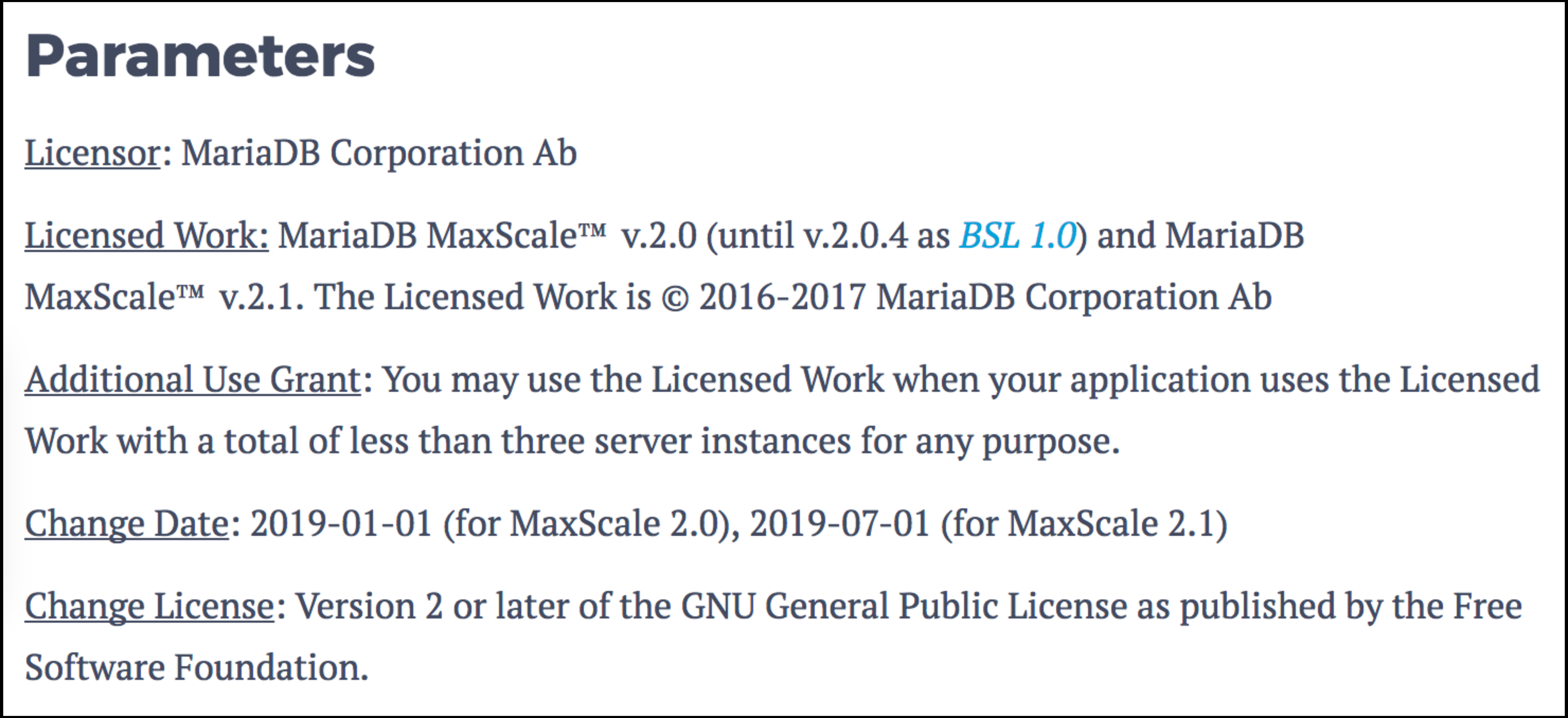


```
Use Limitation: Usage of the software is free when your application uses
                the Software with a total of less than three database
                server instances for production purposes.

Change Date:    2019-01-01

Change License: Version 2 or later of the GNU General Public License as
                published by the Free Software Foundation.

For information about alternative licensing arrangements for the Software,
please visit: https://mariadb.com/products/mariadb-enterprise
```



Parameters

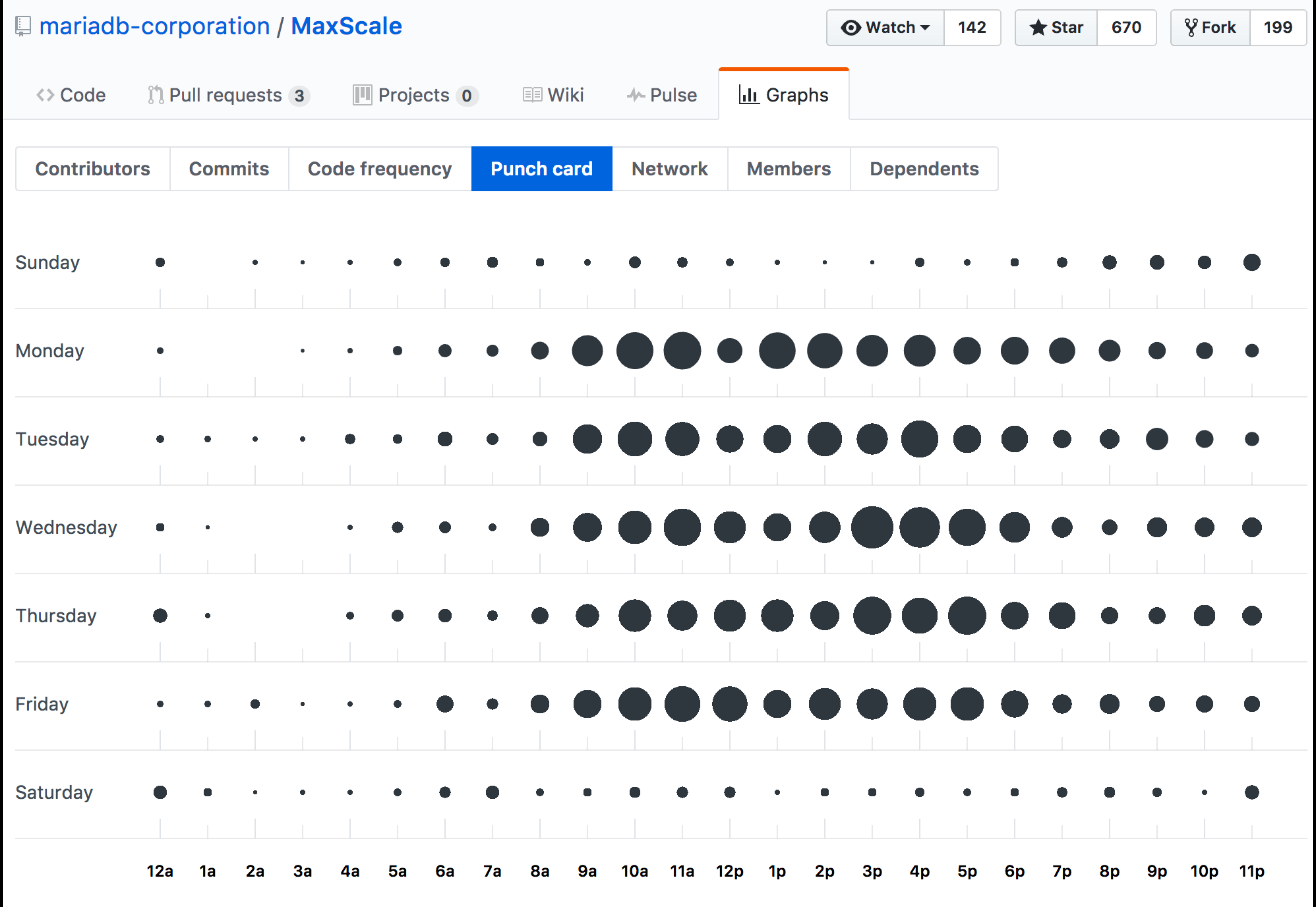
Licensor: MariaDB Corporation Ab

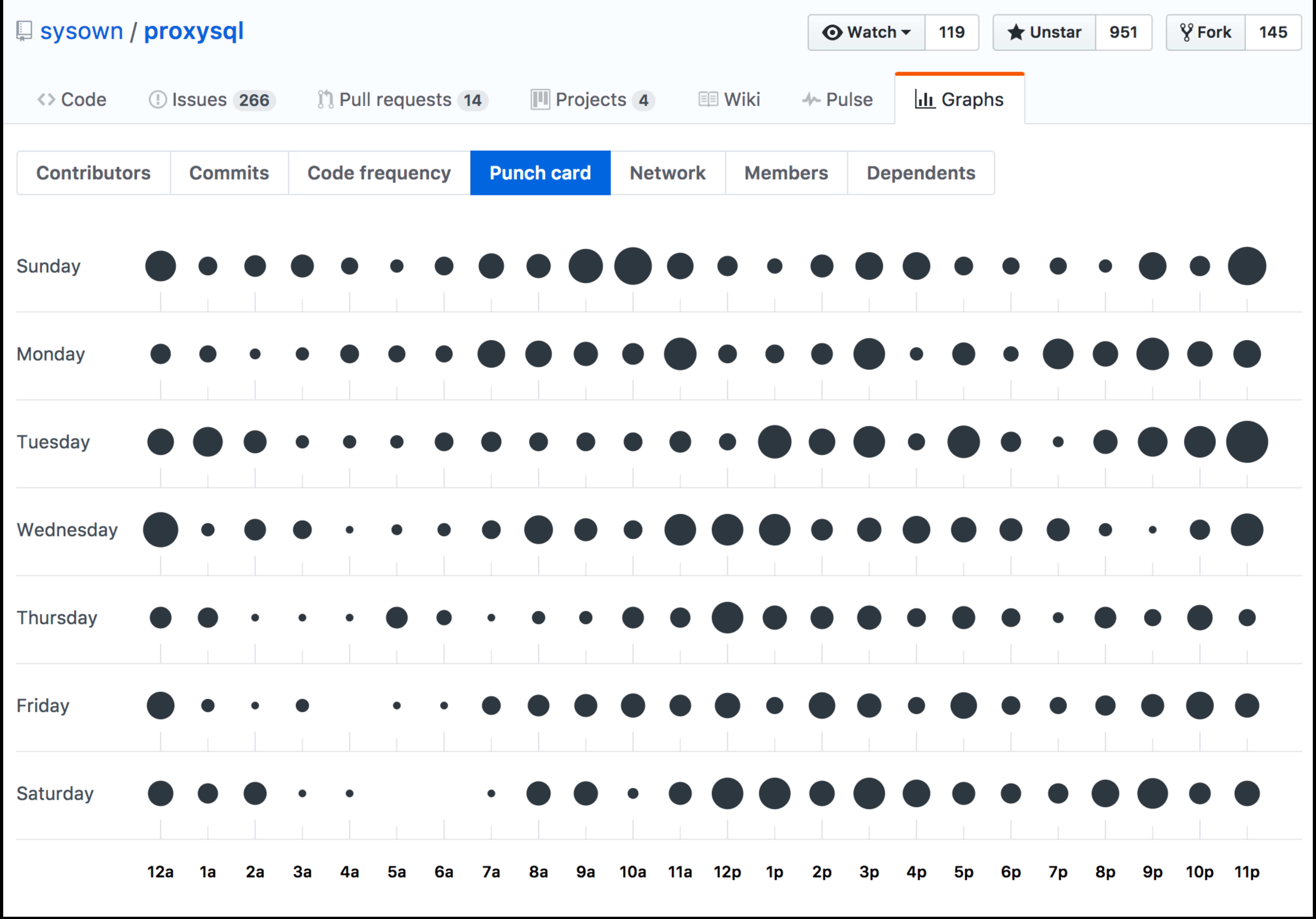
Licensed Work: MariaDB MaxScale™ v.2.0 (until v.2.0.4 as *BSL 1.0*) and MariaDB MaxScale™ v.2.1. The Licensed Work is © 2016-2017 MariaDB Corporation Ab

Additional Use Grant: You may use the Licensed Work when your application uses the Licensed Work with a total of less than three server instances for any purpose.

Change Date: 2019-01-01 (for MaxScale 2.0), 2019-07-01 (for MaxScale 2.1)

Change License: Version 2 or later of the GNU General Public License as published by the Free Software Foundation.


mariadb-corporation / MaxScale
Watch 142
Star 670
Fork 199
Code
Pull requests 3
Projects 0
Wiki
Pulse
Graphs
Contributors
Commits
Code frequency
Punch card
Network
Members
Dependents
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
12a
1a
2a
3a
4a
5a
6a
7a
8a
9a
10a
11a
12p
1p
2p
3p
4p
5p
6p
7p
8p
9p
10p
11p


sysown / proxysql
Watch
119
Unstar
951
Fork
145
Code
Issues 266
Pull requests 14
Projects 4
Wiki
Pulse
Graphs
Contributors
Commits
Code frequency
Punch card
Network
Members
Dependents
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
12a
1a
2a
3a
4a
5a
6a
7a
8a
9a
10a
11a
12p
1p
2p
3p
4p
5p
6p
7p
8p
9p
10p
11p

No forks?

- Well, AirBnB MaxScale was a fork announced before the BSL
- There was another non-credible fork made too by a community member
- Why? Because it wasn’t depended upon by people, and there were already two very good open source alternatives (and ProxySQL is widely utilised)
- What could have been done better? **Early warning vs. shock & awe**

The cloud meets open source software

- Many successful open source projects have alternatives in the cloud

- MySQL is a good example, and it is estimated that Amazon alone brings in more revenue yearly on their RDS MySQL service than all the MySQL ecosystem service providers combined (including Oracle!)

Redis Labs

- Raised $86m, employs the creator of Redis
- Redis itself, is BSD licensed (permissive!). Redis Modules can be licensed as whatever the creator deems
- So some Redis Labs modules are Apache 2.0 + Commons Clause (making it clearly, not Apache licensed)
- Commons Clause = source available software (not new, Microsoft had this with Shared Source back in 2001)

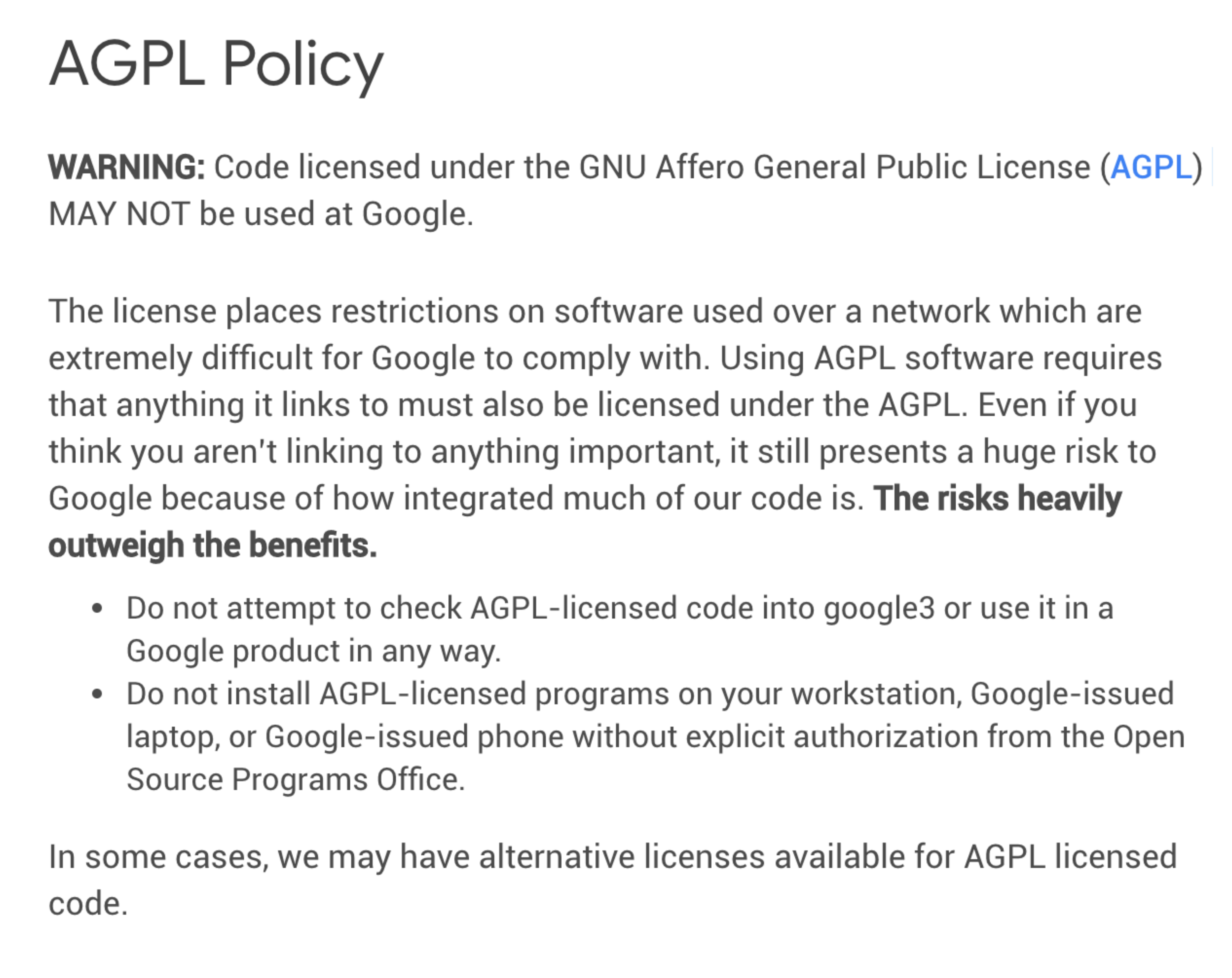
Redis Labs II

- *"Today's cloud providers have repeatedly violated this ethos by taking advantage of successful open source projects and repackaging them into competitive, proprietary service offerings. Cloud providers contribute very little (if anything) to those open source projects. Instead, they use their monopolistic nature to derive hundreds of millions dollars in revenues from them. Already, this behavior has damaged open source communities and put some of the companies that support them out of business."*

- AGPL didn't work for 2 reasons:

  - cloud providers could create managed services

  - large enterprises wanted a more permissive license as using the AGPL was against company policies



AGPL Policy

WARNING: Code licensed under the GNU Affero General Public License (AGPL) MAY NOT be used at Google.

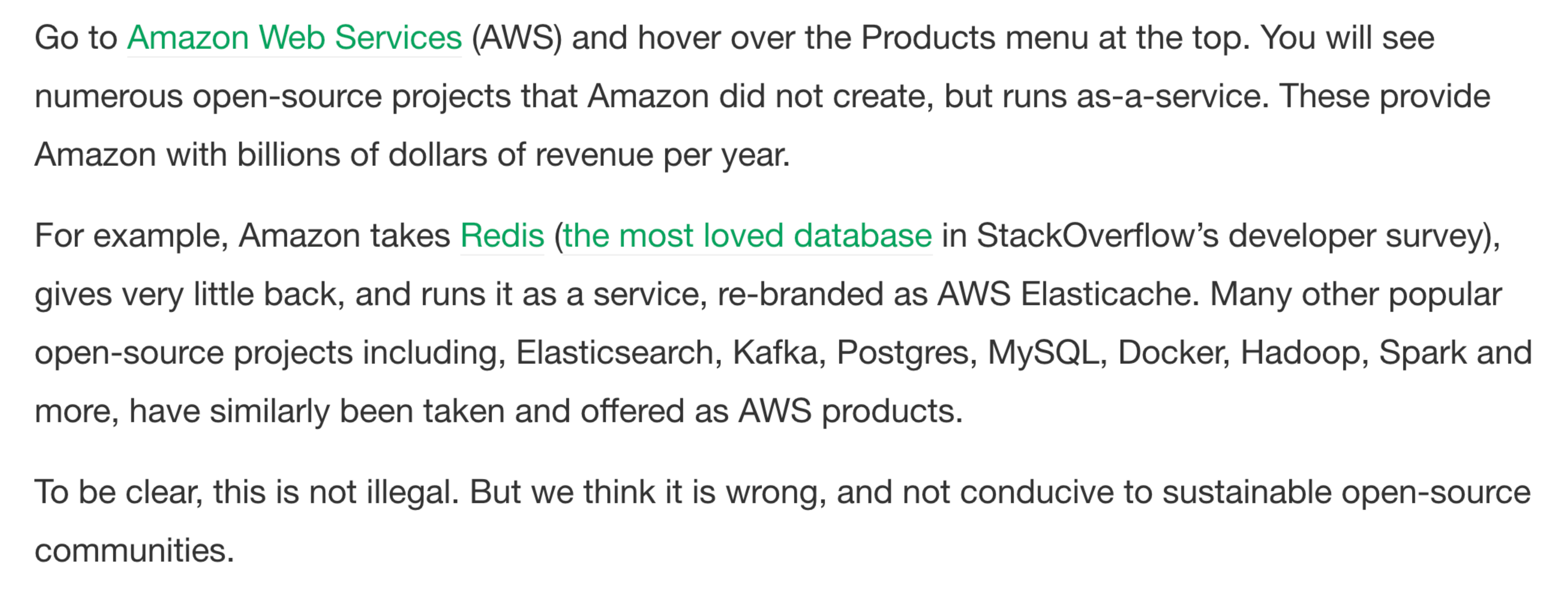
The license places restrictions on software used over a network which are extremely difficult for Google to comply with. Using AGPL software requires that anything it links to must also be licensed under the AGPL. Even if you think you aren't linking to anything important, it still presents a huge risk to Google because of how integrated much of our code is. **The risks heavily outweigh the benefits.**

- Do not attempt to check AGPL-licensed code into google3 or use it in a Google product in any way.
- Do not install AGPL-licensed programs on your workstation, Google-issued laptop, or Google-issued phone without explicit authorization from the Open Source Programs Office.

In some cases, we may have alternative licenses available for AGPL licensed code.

Redis Labs III

- Now, Redis Labs isn't a company that is struggling by any means. Sources via Wikipedia suggest they have over 3,000 paying customers in September 2014, and that was **huge growth after changing their name from Garantia Data to Redis Labs (December 2013: 1,300 paying customers)**. It helps that the **trademarks are in favour of Redis Labs**. More recently, Redis Labs Announced their **10th Consecutive Quarter of Double Digit Growth boasting over 8,200 customers**. Both Forrester Research and Gartner have held Redis Labs in high regard. So the motivation must be the cloud providers making money from Redis Labs work; as Sanfilippo blogged, "it’s not ok to give away that value to everybody willing to resell it. An example of such module is RediSearch: it was AGPL and is now going to be Apache + Common Clause."



Go to Amazon Web Services (AWS) and hover over the Products menu at the top. You will see numerous open-source projects that Amazon did not create, but runs as-a-service. These provide Amazon with billions of dollars of revenue per year.

For example, Amazon takes Redis (the most loved database in StackOverflow’s developer survey), gives very little back, and runs it as a service, re-branded as AWS Elasticache. Many other popular open-source projects including, Elasticsearch, Kafka, Postgres, MySQL, Docker, Hadoop, Spark and more, have similarly been taken and offered as AWS products.

To be clear, this is not illegal. But we think it is wrong, and not conducive to sustainable open-source communities.

Community reactions

- Positively via a fork of the modules:
  - https://goodformcode.com/
  - Done via Debian and Fedora maintainers

MongoDB

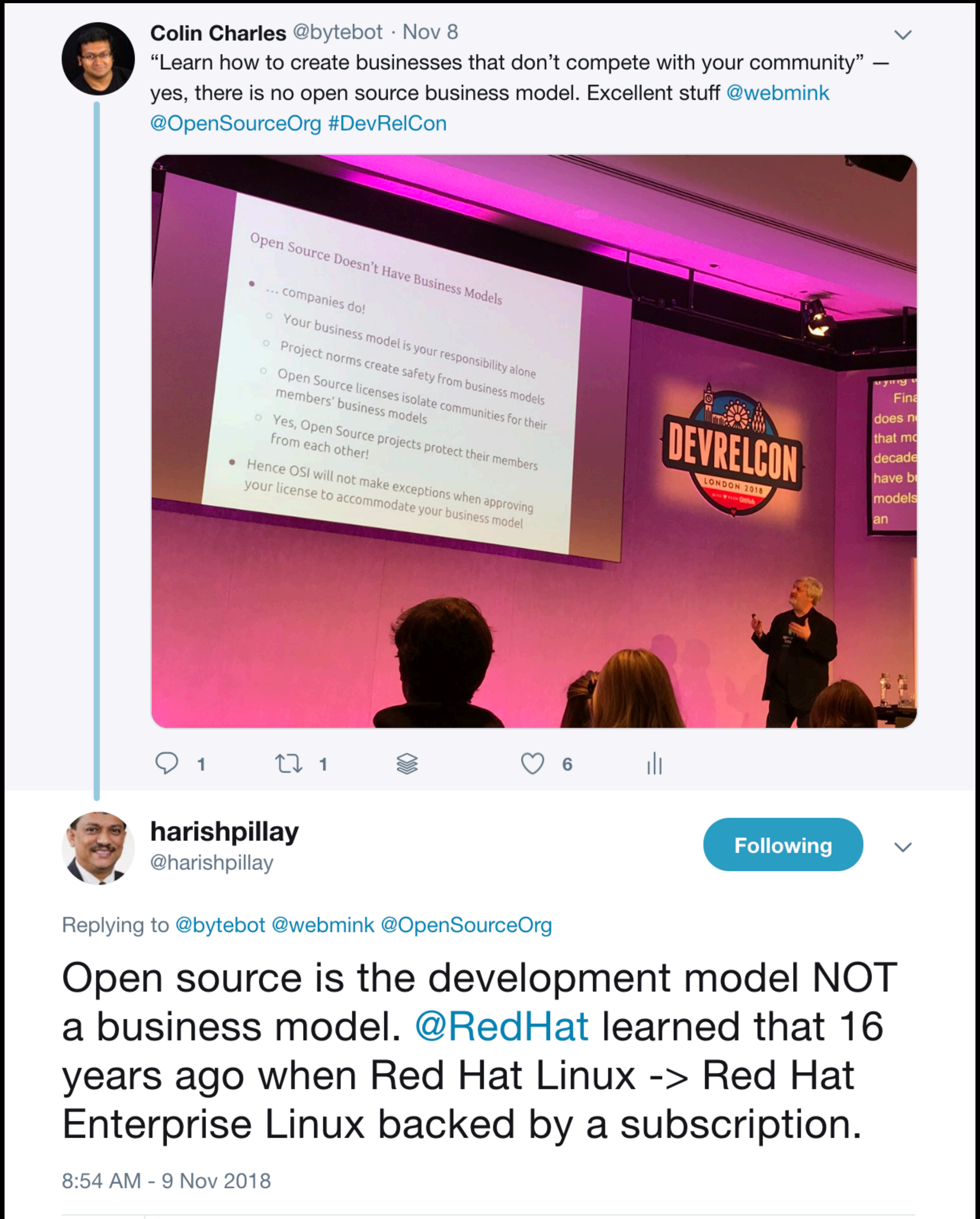
- Built an amazing moat (server: AGPLv3, connectors: Apache 2) with Enterprise, Stitch, Mobile, Charts, Ops Manager, Compass, Atlas
    - this is what a successful open source company looks like! SaaS that others find hard to replicate
- Took the company public in 2018
- But decided to announce Server Side Public License (and aim for OSI approval)

What happens next…

- There is active discussions on the license at this very moment
- No one has decided to fork the software, but all this is very new
- https://www.mongodb.com/licensing/server-side-public-license/faq

Using contributor license agreements for fun & profit

Some open-core products require their contributors to sign a contributor license agreement, which either dictates that the copyright of all contributions to the product become the property of its owner, or that the product's owner is given an unlimited, non-exclusive license to use the contributions, but the authors retain copyright ownership. In an open-core scenario, these agreements are typically meant to **allow the commercial owner of the product** (which in some cases, is ultimately the copyright holder to all of its code, regardless of its original author) **to simultaneously market versions of the product under open-source and non-free licenses**. This is in contrast with more traditional uses of CLAs, which are meant solely to allow the steward of an open-source project to defend its copyright, or guarantee that the code will only ever be made available under open-source terms, thus protecting it from becoming open core.


Open Source Doesn't Have Business Models
... companies do!
Your business model is your responsibility alone
Project norms create safety from business models
Open Source licenses isolate communities for their members' business models
Yes, Open Source projects protect their members from each other!
Hence OSI will not make exceptions when approving your license to accommodate your business model
DEVRELCON
LONDON 2018

Colin Charles @bytebot · Nov 8

“Learn how to create businesses that don’t compete with your community” — yes, there is no open source business model. Excellent stuff @webmink @OpenSourceOrg #DevRelCon

harishpillay
@harishpillay

Replying to @bytebot @webmink @OpenSourceOrg

Open source is the development model NOT a business model. @RedHat learned that 16 years ago when Red Hat Linux -> Red Hat Enterprise Linux backed by a subscription.

8:54 AM - 9 Nov 2018

So, how do you make money with open source?

- Subscriptions!
- Dual licensing
- Pfessional services
- Software as a Service is a great model (think tools in a web browser even)
- Proprietary plugins
- Time delayed open source
- If you can, advertising partnerships (see: Mozilla, Ubuntu)

But… how do you make money and not piss your community off?

- Do not give them the shock & awe treatment
- Do respect the community since they are likely what brought you success in the first place
- Provide real differentiation or additional features that people may be able to provide alternatives to if required
- Enable your community to provide good, valuable feedback before you plan to make changes (learn from the hospitality industry and loyalty points)

Thank You!

Colin Charles

colin@grokopen.com | @bytebot on twitter