EPISODE 895

[INTRODUCTION]

[00:00:00] JM: Open source software has evolved into a thriving multifaceted ecosystem. Open source encompasses operating systems and databases. Open source embodies both altruism and self-interest, and open source enables thriving businesses from WordPress blogs to hundred billion cloud providers. There's a large set of business models that can be built around a successful open source project. Some of these business models are more defensible than others. A company based around open source must make a deliberate set of tradeoffs around how their private company will relate to the public open source ecosystem.

Today's episode is a discussion of open source business models with Karthik Ranganathan, Heather Meeker and Matt Asay. Karthik is the CTO of YugaByte, an open source distributed SQL database. Heather Meeker is an open-source licensing specialist and a founding partner of OSS Capital, and Matt Asay is a longtime executive who has worked in several open source based companies, including MongoDB. He’s also written at length about open source.

Full disclosure, YugaByte, where Karthik works, is a sponsor of Software Engineering Daily.

[SPONSOR MESSAGE]

[00:01:24] JM: SQL has been around for a very long time, and the basics of SQL might not change very much from year-to-year, but the underlying technology that implements those queries is undergoing constant innovation. The Distributed SQL Summit is a full day of talks about building and scaling distributed SQL systems in the cloud.

The Distributed SQL Summit is September 20th, 2019 in San Jose, California. Distributed SQL databases can globally distribute data and elastically scale while also delivering strong consistency and ACID transactions.

The Distributed SQL Summit includes speakers from Google Spanner, Amazon Aurora Facebook, Pivotal and YugaByteDB. To find out about the latest innovations in large scale distributed systems infrastructure, mostly with a focus on distributed databases, check out the Distributed SQL Summit, September 20th, 2019 in San Jose.

[INTERVIEW]

[00:02:34] JM: Matt, Heather, and Karthik. Guys, welcome to Software Engineering Daily .

[00:02:38] HM: Thank you.

[00:02:39] KR: Thank you.

[00:02:39] MA: Thanks. Thanks for having us.

[00:02:41] JM: The subject of today's conversation will be open source business models, and I'd like to start with a bit of history in the world of open source. To my mind, the first big example of a successful open source company was Red Hat, and I'd like to start with just each of your reflections on the early open source business models including Red Hat. Maybe could start with Heather.

[00:03:06] HM: Sure. Well, of course Red Hat is a great example that everybody is familiar with, and interesting – I'm sure we'll get into this more, but interesting because it's sometimes referred to as a pure open source business model. So open source software selling services and quality assurance, right?

Another very early one that's probably worth mentioning is MySQL. That also was in early days and was a different kind of business model. So, very interesting. I mean, back then, people used to ask me all the time, "ow can you possibly make money selling open source software?" and of course I've been a lawyer in Silicon Valley for many years. So my answer was always, "I'm a lawyer in Silicon Valley. None of my clients make money," but, nevertheless, there are ways to make money, and it's interesting that those two early examples went about it in a fairly different way. But those are the ones that really, really stick out in my mind as early days.

[00:04:14] JM: Karthik?

[00:04:15] KR: Sure. Yeah, I'll give you a technologist's point of view life. For me, back in the day, I mean, like I was experimenting with Linux early on. Trying to get things like the mouse working, which was incredibly hard, and i you ask people back then, nobody would have ever thought that something written by a bunch of people for free in their supposed spare time could ever become a business or even a viable technology tool. But yet here we are, open source is becoming more mainstream.

Red Hat is doing really well and everybody uses Linux. So I think that's a great testament to say that the early days, there was more confusion about whether things would work out or how they would proceed, but now it's becoming more mainstream at least in the operating systems and Linux side of things and specifically with Red Hat as a monetizing component as a business model on top of that.

[00:05:01] JM: All right. Matt, have about you?

[00:05:03] MA: What's interesting is that, to what Heather said about Red Hat, being the first pure play "open-source company?" What's really interesting to that is that no one or – Well, for years, no one seemed to follow them. So they have this model that seemed to work, seemed to work well, seemed to work in certain kinds of situations where you had to have a fair amount of complexity that you could then charge for simplifying or packaging up in a distribution and selling that access to that distribution.

But until – I don't know, Hortonworks, and I'm trying to think of some of the others, more recent examples. No one espoused that same business model despite the fact that it was the one example of an open-source model that seemed to work at scale first company to a billion.

I worked for a range of companies that tried different iterations on – I was going to say Red Hat model, but really we didn't try the Red Had model. What we tried were different ways to make open source look less like open source and more like the proprietary package software business that we had come from.

So you’d have things like community version with an enterprise version, and that enterprise version was sold as a proprietary license under a proprietary license and that’s – Called the open core model. That became the most prevalent model, again, which is a little bit weird given that the one company that showed that you could make a ton of money in open source was Red Hat going for, again, this pure play model. People can quibble about whether how pure it was, but a relatively pure model. Since then, we've had various iterations on open core, but not so much on that original Red Hat model of 100% open source.

[00:07:02] HM: If I could make a comment there. One of the interesting things about that I think is that because Red Hat was Linux, and by far the most successful business at Linux, one of the reasons that people were wondering whether that could be replicated elsewhere as supposed to open core, which was more the route people were taking for different kinds of software, is that Red Hat had this enormous product base, because it was dealing with Linux, which was the absolute killer app of free software.

In other businesses, it was not quite as obvious that you could do things at a tremendous scale, like Red Hat. Now, I think people are starting to moderate their views on that, but for a long-time people would say something like there's only one Red Hat. What they meant by that was not just in the Linux space, but that that business model was not necessarily what people expected to replicate in other sort of software markets.

[00:08:12] JM: So there are many different areas of open source business that we could explore, and when I think about the timeline of milestones from those early days of Red Hat and perhaps MySQL, to today, I think about companies like Spring, and Spring Source, and eventually Pivotal which where you kind of a services model around core open-source business, and then you have some other adjacencies that come out of the services plus open source model. We've also seen the rise of kind of the hosted cloud offering. We've seen the rise of the Hadoop vendors, but I think the area that I'd like this conversation to focus on for a variety of reasons that will become emergent is open source database models.

When I say database, I'm using that term quite loosely. I'm talking about both database companies like MongoDB and CockroachDB, but also these things that are essentially databases, like Elasticsearch, like a big search index is basically a database. Confluent, which is a Kafka provider, which is Kafka is essentially a big distributed queue, but that's a database. Redis, which is an in-memory object caching system, but you could call it a database. The reason I'd like the conversation to focus on these types of companies is both because there's a lot of debate around licensing issues in these companies, and also because Karthik is running a database company, so he can get some firsthand experience on this.

So as we get into the discussion of database companies, modern open source database companies, if you're building a database company, what are the options to monetize that database? Probably, we could start with Karthik, because you're actually in charge of a database company.

[00:10:05] KR: Absolutely. Sure, Jeff. So here's the way, at least over the three years. Our view has been shaped and reshaped multiple times, and obviously as a startup, you have to adapt and keep learning, right? We started the company, and I can just go through it as we observed it as a company, as the journey happened for me as it unfolded.

Starting the company in 2016 was very interesting, because there was a lot of skepticism on if open source would work out as a business model at all, and a lot of people cautioned us, "Stay away from it. Be very careful," and open-source being a one-way street, as in once you've open sourced, you have open sourced, but they told us like, "Take your time and think through it."

So we thought to it. We took about a year. We went and studied a lot of businesses. A lot of ways how they were like gaining adoption, gaining popularity, offering value, what the end customer and the user behavior was, so on so forth, in this space. Then fast-forward to 2017 when we were pretty clear we wanted to open-source, but the million dollar question in front of us was do we do AGPL, which at that time was fondly referred to as the Amazon GPL, so that it offered you protection against Amazon, or do we do Apache 2.0? This was another fork in the road where a lot of people say, "Don't put it open. They'll take it and run it," yadi-yada, etc., etc.

What we said was, "Okay, we're going to take a moderate stance at that point," because again things were not fully clear and we said we are going to go Apache 2.0 for a majority of our core database with some portion held off in the enterprise edition, which is not even open source. So is just value-added stuff. This refers to what Matt was alluding to, where there are companies that are trying to imitate enterprises. So that's how we went about it.

But now, come 2018, ’19, as time unfurled, what we’ve realized was that in seeking to monetize it prematurely and in seeking to monetize everything, we are actually giving up a much bigger opportunity and an opportunity which is almost akin to making a revolution in the database space.

So our current viewpoint is that there is enough proof in the pudding that open-source does not detract from monetization. It most likely means if you couldn't have monetized some of these effectively anyway, that's what it means if people don't pay you, but adapt to your open source, and cloud and operational experience is the place to monetize.

Again, I will echo what both Heather and Matt said about Red Hat being the only business and etc., etc. But the other thing that's become painfully evident is that the more mission-critical the infrastructure piece, like the data infrastructure piece, the more important it is both for it to be fully open so that users can build their trust and be sure this is something that will service a large number of people, but also will pay for the support and whatever comes afterward for the convenience of using it in production.

So if you’re putting that together, that's our viewpoint. Make the database 100% open, but the important things for – Like other people thinking in the same shoes that we would say are, “Make sure it's mission-critical and make sure it's a large market. Those two are very important.”

[00:13:02] MA: Actually, if I could jump in on that to add to what Karthik said, Jerry Brostoff, a former colleague of mine at MongoDB wrote this great post on the secret to making money with open source. He boiled it down to exactly what Karthik just said, that customers, enterprises, will pay for software that runs their mission-critical applications. So if they go into production on something and they come to depend on that, then they're going to pay for the operating systems. They’re going to pay for the database. They're going to pay for that infrastructure that's running that application.

Frankly, some of the problems that I've had in the past with different companies like, say, at Alfresco, open source content management system. A lot of the use cases that people were running in Alfresco for weren’t really mission-critical. They were maybe departmental applications or otherwise, but there wasn't a compelling reason for them to pay. So we came up with these licensing gymnastics. Went through these licensing gymnastics to try to get them to pay, but it missed the point of the reason that they weren't paying wasn't so much because of the license that we had. It had everything to do with how critical was our software to running their applications.

One of the great things, I remember having this conversation with Max Schireson, the former CEO of MongoDB when I joined, or actually when I was going to join, and I was asking him, “Hey, what are you going to do about this payment problem that I've had over and over in open source?” He said, “Matt, there's nothing stickier, nothing harder, nothing more important than the database. Therefore people will pay.”

That may be a slight overstatement, but it's borne out in practice. MongoDB has done really well in the market and it has really nothing to do with their licensing and everything to do with as companies build on their product build on a database. They're going to pay to make sure that it doesn't go down, or if it does go down, they can get support, they can get somebody on the phone to fix it.

[00:15:20] HM: I would also build on what Matt was saying and what Karthik was saying. I think that at this point, we're in transition state, and in a way we’re always in a transition state in technology, but that critical infrastructure is moving up the stack. So while we may have considered operating systems, the critical infrastructure over the last 20 years. Now, things like database and other sorts of utilities are moving up into the critical infrastructure space.

So as the technology advances, the business and licensing models advance too. So it's not at all surprising that there would be transitional steps in moving from something that's more like open core or more like proprietary or more like a services model to more of a “pure open source model”, because as the product that you're selling gets ubiquitous, and to some extent, commoditize. What people are paying for is quality control, a little bit more than they're paying for like the latest feature, because people are tending to cooperate on that. But I am curious whether the rest of you agree with that or not.

[00:16:53] JM: Real quick. I just want to say, you make a profound point there, and I think this is maybe – It’s one of those points where it’s obvious after I hear it. But just the idea that the operating system is basically we've settled on Linux. That's what everybody's using. I mean, even Microsoft is using Linux at this point. So everybody's using Linux. So what is the operating environment that we’re focused on? It’s something higher-level. It's the database, or perhaps the orchestration system, or something else. But you can think of the database as almost the lowest level operational element that we are working with.

Just to move the conversation forward. Heather, I would like to get some context from you on why open source licenses have become a point of debate. Catch people up for those who have not been paying attention to the licensing discussions, or maybe they've heard it from the edges, but they haven't really paid close enough attention. Why are open source licenses becoming a point of debate?

[00:17:58] HM: Well, I would say they are a point of debate mostly – It's mostly an inside poo thing, right? When you look at the debate that's going on, I think you have to sort of weave out what debate is taking place among people for like focused on these issues every day versus sort of the rest of the world who is may be actually adding a little bit more value to the technology landscape. I say that about myself because I'm a lawyer. So there are a lot of people who are sort of focused on modeling and so forth, but in a sense I think we all should understand people in my position that it's other people who are actually creating the software and the value.

There are a few things that are the subject of debate these days. One thing that's the subject of debate is – And we saw this with the license that Mongo recently published. How far does Copyleft or the sort of free software sharing model go? How far can it go under the law? How far should it go as a matter of community rules?

For many years, we’ve had GPL, which is essentially you share if you distribute binaries. You share a source code. Then we had AGPL, but that was only well you share if you're using the software, essentially, for SaaS and you've modified it. But that model was not quite as robust. There weren't as many people using it and it didn't become mainstream in quite the way that GPL, the sort of distribution model became.

So when Mongo created this new license, they wanted to go further and try to incentivize people to share other code that they're using that wouldn't be within the ambit of AGPL. So that one argument or discussion that's going on. I would call that fairly insightful, actually. I would say that to most people, that's not a sea change in what's going on with open source.

Another thing that's going on with open source though, I'd say there are two other things that are maybe more relevant to most people. One is that there's a sort of philosophical debate about whether there should be a thing called an open source business. In other words, is it within the spirit of open source to do business at all, or should it all be developed on a sort of nonprofit community basis? One step over from that, maybe we say pure business models like Red Hat are okay, but everything that's open core or services model or whatever isn't. So that's kind of a philosophical debate, and you do have some people saying you shouldn't call yourself an open source business if you're not a pure open source model.

That's the kind of thing that should matter to entrepreneurs who are in the trenches, because if nothing else, it can be a big distraction from what you're doing. When you release a great product, you don't want to be involved in a Twitter storm over whether you're correctly calling yourself an open source business. That's just a distraction, okay? No matter how you come out on the issue.

Then another thing that's been happening, and I'll only describe this briefly, is that there is a new category of license out there and it doesn't have one specific name, but I'll call it a source available license. All of the companies, almost all the companies you've mentioned in the database space, at least for part of their code are using source available licenses. So these are not open source licenses, because they have license limitations.

The different companies you mentioned each has their own, basically, and that's another thing that is getting discussed. Although in a way it's a bit of a strawman, because those aren't open source, they're source available. So what I think is really happening there is this other category is being developed and it's clearly being demanded by the marketplace, the developers who want to license their stuff. But there is philosophical and I guess kind of a branding, naming kind of debate about what we call those things and what their place is alongside open source, say open cores and other open source software.

[00:22:50] JM: Matt, you've been writing about open source for 12 years. Can you give us some historical context for the modern debate about licenses and, more broadly, what open source actually means, because a lot of this debate I think is – Some of it boils down to a kind of fundamentalism, where people are wondering what actually does open source mean. We have certain norms. We have certain expectations, but what are those actually grounded in?

[00:23:26] MA: This is when I start to realize that I'm old. Whenever I think that I'm old until I have one of these conversations, and I say, "You know what? We've had this. We've already discussed this. We already settled it 10 years ago," and then I realize when those words come out of my mouth I'm like, "Oh, wait. I've had this name come [inaudible 00:23:43] over and over [inaudible 00:23:45].

[00:23:46] HM: Those people weren't around then.

[00:23:49] MA: Oh, crap! I'm old. I've become that guy. Get off my lawn. So first of all, I would say that what we're going through right now, I don't think really that it's any different from what we've gone through before. The fear of the public clouds is very similar, if not the exact same as what we had before of the fear of the big web companies. It was Yahoo, if you remember them, Google, etc. The exact same concerns that they were taking software and not contributing back.

So that's always kind of been a concern. Even before the web companies, well, at one point, we were just happy that anybody downloaded software and used it and we're just excited about that. Then as soon as we got over that initial euphoria and said, "Hey, wait. Why aren't you contributing back?" This conversation has been going on forever.

Early on, I was not part of this conversation, but early on. What was it? '97? They came up with the open source definition. Heather, you'd be a better expert on this, but either it came – I know they've termed open source, it was coined in '97. I don't know if the open source definition came out at the same time. But at some point in the distant past, open source was defined. For me, the definition is – And you can go to OSIs webpage or just type in open source definition into Google and you'll find it. For me, that's the canonical definition that we've been following more or less. When you say open-source, it's roughly what everyone thinks of when they hear those words.

The difference has been that there are always companies, and I've been a part of those companies and I've helped drive those business strategies for those companies. There are always companies who were trying to get the benefits of open source without necessarily – I was going to say, and are surprised than when people want people like an Amazon or a Microsoft or somebody else comes in and says, "I would like some of those benefits too. I'll take your code and I won't give anything back, because I don't have to."

People seemed surprised that they can't just have the benefits of free distribution, marketing, etc., and not have people actually show up and use the software. But, again, these are not new. They're not new discussions. They're not new problems. The one thing that's potentially new is when open source started way back when, decades ago, I think that there were very few people who were actually involved and the people that were downloading Linux, I would hazard a guess, they knew something about the license and they cared about license.

Fast-forward to today, however many years that is, and I don't think many developers, and this kind of came out and what Heather was saying earlier. I actually don't think most developers most of the time are thinking about the license. I call it the GitHub generation, because years ago, GitHub woke up and discovered that most of the repositories didn't carry a license at all. So the developers were saying, "Hey, use this," and Heather as a lawyer, I'm a fake lawyer, Heather's a real lawyer, will know that you don't just – There's always a copyright or something attached to that software even if those developers didn't intentionally apply one. There was something there. So you have this messiness were developers were like, "Yeah! Whatever. Just go ahead and use it," without applying a license.

So we got this weird in-between world where – And James, governor or RedMonk, called it the post open source world, where developers didn't care about the licenses at all. I think that's where we are today, like for this debate. It's a very fierce debate on Twitter amongst – Again, Heather kind of referenced this. Inside baseball sort of crew of people battling it out over whether this license is actually open source or not, and I think the developers in the background are just still clicking on that repository, pulling down that software and using it and not too bothered by the licenses. Maybe they should be, but I don't think that many developers are thinking about the licenses most of the time.

Now, there are legal departments when we pull software, and we've had this at Adobe. Is that AGPL kind of hands-off? Is that SSPL, that's not open source. We need to talk about this. Is that Apache license? No problem. Legal departments to the extent that they know about the software that's coming into their enterprises are concerned with the licenses, but I don't think developers as a general rule are.

[00:28:40] JM: just to help me understand this. The way that I perceive this debate is ultimately the idea is, in the case of all five of the companies I'm about to mention, MongoDB, Elastic, Confluent, Redis Labs and Cockroach Labs, which makes CockroachDB. In each of these cases, they have made some change to their license policy. Ultimately, the change is to allow their company to have more share of the profits that are being earned by the open source project that their company's resources are maintaining, relative to, specifically, Amazon Web Services, because Amazon Web Services has a hosted proprietary version of each of these five products. Heather, could you just tell me if I'm wrong or add any kind of nuance to what I just said.

[00:29:43] HM: Sure, and I think in this case, a lot of it is about the nuance. When you read stuff on the web and tweets and so forth, a lot of this gets lost. By the way, most of those companies were open core before and are still open core. Not every single one of them.

So what has happened in several of those cases is it that the line between the core and the frosting or some people call it the crust.

[00:30:14] JM: Define those things. Define the open core and the crust.

[00:30:17] HM: Yeah. So open core is some open source software that forms a core functionality of the product, and then some extra bells and whistles, frosting, whatever, which there’s no one named for. I’ll just call it frosting. That is not under an open source license. There are many variations on the model, right? There are sometimes three categories instead of two and so far and so on.

But if you look at it from a really high level view, most of those companies were open core before and they’re open core after, but they have adjusted the bucket. So software that they’re making available. So they moved software from the open core into the frosting or vice versa, right? So that's one thing to point out, is that, in a way, some of these licensing changes are not as radical as you might think from what you read about them.

The other thing I would say is that while this is normally discussed as an Amazon issue, it's not just an Amazon issue. The same rules apply for all of the big cloud providers and it really does come down to a pretty simple question, which is these are all private companies. They are developing with private capital. The question is how much are they willing to leave on the table and allow the big cloud companies to make money from their software without the developer monetizing any of that.

I’d pointed out, as I always do, that when somebody free rides on your software, which is by the way one of the points of open source, right? It doesn't mean you're losing money. It just means there's an opportunity cost, and each of these companies have adjusted their licensing models in ways that suit their own preferences for how much they're willing to leave on the table.

[00:32:21] JM: Okay. So what has been your reaction to the license changes? I think we should take each of you in isolations. Maybe we can start with Karthik. What has been your reaction as somebody who's been a developer in the open source and cloud ecosystem for a long time? What's been your reaction to the licensing changes?

[00:32:44] KR: Yeah, a great question. So I'll give you two perspectives, and all the companies you mentioned, although they are grouped similarly, actually have nuances that make them different. So there are different buckets in what the changes are and so on and so forth.

Firstly, as Matt said, if I were just putting a developer hat on, I wouldn't care. If I could use the product, I would, and the more of the product that I could use, the happier I am. That's simple.

But now if I am the developer in a company building software for a company, or more like in the position of a team lead or a tech lead or an architect making a decision for a product, I start the care, because I don't want my project to get shut down after it was successfully almost ready to get deployed, but the legal team comes in and says, “You know what? This is dangerous. This will put the business at risk. Don't do this.” Therefore, I start to care.

Now, if you take a slightly longer term view still as an architect or a decision-maker and as a technologist, I start to care because the more open your ecosystem, the more competitive. Yes, that's bad on the monetary side, but it's great for the product. It keeps you honest and it keeps you building the very features that your users and your community demand. So that means you’re servicing them the best and you’re building something that's entirely defensible in terms of its merit as a product, which is the primary purpose of this whole thing anywhere. If there was no product, there's no need to be open sourcing or having any of these discussions.

So that aside, if you look at the changes that have happened, there's actually a few different buckets, right/ There was the open core portion and there was a portion of the crust, or the frosting as Heather put it, that people often monetize on. A lot of the changes, specifically MongoDB and Cockroach, have moved their core into a different license, which effectively makes the project itself noncompetitive.

So that means, yes, it will hurt Amazon and it prevent them from putting the project as it is into production. But it’s not just Amazon in Mongo's case. It's Azure, Azure Cosmos DB also offers a Mongo API. So at this point it begs the question; did the change in license really prevent anything? Well, it didn't hurt the big cloud providers, but it definitely hurt all of the other people who are the smaller hosting providers who might have arguably contributed to the ecosystem itself. That is what made the Mongo community itself. So that's one piece.

As Heather said, it is private capital and you know companies have to make money. So the move is not illegal. It’s something they can do, but it definitely sparks a debate, because the smaller hosting providers find themselves, the same move done on them that the large cloud providers would do to the largest, most successful projects. So that’s on one side.

. If you take companies like Confluent and Elastic, I think they have merely secured their enterprise frosting, the value-added portion, but made its source available. So that means that the core is still open and competitive, and that's a good thing. Yes, is it necessary? Is it going to do anything? Well, in Elastic's case, they put all of their security features out as source available, which means it's not an open license. But since Elastic was very popular, Amazon rebuilt those and open sourced it, at which point Elastic had to open source it also. So the mileage may vary, but it definitely buys you some time.

Now, that brings us to Cockroach Labs and ourselves, YubaByte. We're in a very similar situation and are actually very different from these other projects. If you think about what Mongo, Confluent, Elastic and Redis do or did, they came up with an entirely new API, completely new. So they had to evangelize and teach developers how to think and build in those APIs.

YugaByte and CockroachLabs, however, we build on existing APIs and enhance them. That's because there are so many database APIs out there. Unless we're really doing something fundamental, we don't want to introduce a new API. So in YugaByte's and Cockroach's case, we both used the Postgres API.

Now, Amazon already has a Postgres compliant database called Aurora. People might've heard of it. It's doing well over a billion dollars in annual run rate. So I don't think Amazon is really going to take any of the smaller projects that are evolving at this time and to offer them as a service right away, firstly. Secondly, keeping the core open actually makes us as vendors and developers of software more honest, because, yes, there's more competition, but that's what will force us to prioritize and figure out what is the value-add we can bring to the user. So there are my two perspectives as a pure developer and as somebody building software and running a company at this point.

[00:37:30] JM: And, Matt, how about you? How did you respond to these license changes with your historical perspective and your past writing on this topic?

[]00:37:41 MA: Well, maybe I'll speak to the MongoDb example in most detail, because that's the one I'm most familiar with, since I used to work there. Frankly, I was really frustrated, because I understood to why they were doing it. It was this perception that they had to protect their primary revenue source from the public cloud providers, so some irony in there.

The stated cloud providers that they were trying to hedge against were the Chinese cloud providers. I know with some certainty that the Chinese cloud providers when presented with a new license said, “We don't think that we have to abide by that either.”

So it didn't work is intended for that crew. Maybe it worked as – It wasn't necessary with regard to the Amazons of the world, because Amazon and Microsoft and Google were already not using the MongoDB software in part probably because of the AGPL. But I think that the bigger point that I would make in it is that all of these – Again, I’ll say the licensing gymnastics, are not counter to, but ignore the real problem, which is how – Again, if we go back to mission – Why do customers pay? Why do enterprises pay?

They pay if they’re running mission-critical applications on it. The license that MongoDB or Cockroach or somebody else has, has nothing to do with whether – Actually, it may have a negative impact on whether it gets used in a mission-critical application or not because, because if it’s hard to use, if you have to – If you’re going to force them to pay up front, etc., you may never actually make it to that point where they've installed and are actively using the software in production such that they know that they have a problem and they need to pay for it.

But I would say that the thing that Mongo has done and Elastic, to a lesser extent, but they're also doing this, is they've started cloudefying or distributing their software as a cloud service, and that in the case of MongDB, it’s Atlas, which is now I think almost 40% of their revenue and it's not just a matter of cannibalizing their existing revenue. It's actually expanded the pie and it's doing what the public clouds, what they were worried the public clouds would do, which was they’re going to come in and they are going to operationalize our software better than we have.

Well, Mongo has shown that they can operationalize their software and provide customer value at least as well as the public clouds, and it's paying off big time for them. Again, once they've done that, it really has nothing to do with license anymore. It's everything to do with making the life of the developer, her life at whatever company she works at, making it better, making her more productive. That's the key to revenue. It's not about which license that they use.

Heather may disagree with that. But, ultimately, it's about the value that they're providing the customer. I don't want to say that all of their revenue worries go away if they just figure out how to satisfy the customer, because of course even customers who are satisfied running mission-critical applications. If they can get away with not paying for something, of course, they will. It's a rational act.

But in the case of running the software as a cloud service, which to me is the key to monetizing open source forever. If you can do that, if your software lends itself to being run as a service in the cloud and you can do that well, then you don't have to worry about all the problems with licensing. Because, theoretically, if you're the source of the code, you should always be the best option for running it as a service, for providing supports, etc., without a license to insist on that fact.

[00:41:48] JM: Right. So I just want to pause on this point, because I think this is what takes this whole question or debate or Twitter-diverse thing so peculiar, is it's like a red herring. This licensing thing is like almost orthogonal to what actually is going to make business difference in the long run.

So MongoDB's hosted Atlas Service is something that I don't believe AWS has a competitive service to, and is something that's very difficult to build a competitive service to. It's very reliant on UX, UI. There're additional things they can build around it. There's plenty of room to expand that serverless offering. I think with Elastic, there's something very similar, where like if you just think about the idea of an enterprise search product. I mean, look at Google, right? Google is generalized consumer search. We're still finding ways to improve generalized consumer search. So it's going to be plenty of opportunities as a search company to improve your core product, to differentiate from the commodity hosted cloud service, and there's going to plenty of market share to capture.

So, to me, it's perplexing that you would even like kind of come to the table of the whole licensing change sort of thing and rock the boat on the open source ecosystem. Maybe these ecosystems are just different and they vary case-by-case, but it's like the same thing goes for

Confluent. I think I've heard Confluent’s license change was much even more subtle. I don’t know the details there.

But like Kafka is another product that has so much depth to it that it's like you're not really – I don't think you're really competing with AWS. AWS can't compete with you. It's too much of a specialty one-off product. So that said, Heather, I guess I want to get – I also want to get your thoughts on kind of the interplay between business model and licensing before we move on from this topic.

[00:44:09] HM: Well, yeah. So I'll start by saying that I wrote almost all the licenses you are talking about. Now I'm going to say something typically that –

[00:44:22] JM: I hope you blister me. I hope you'd like totally prove me wrong, because I've been ranting on this for too much to know so little about it.

[00:44:26] HM: The thing is people don’t buy licenses, like exactly what Matt said, right? People buy products. Why do they buy products? Because they solve a problem and make life easier. Nobody ever bought a product because of a license. They may buy a product, not buy a product because of the license. But, really, in the long run, what the licensing does is – There is one important thing that it does, and then I'll say what it doesn't do. It does help some developers solve particularly some short term market inefficiencies when people are using their software, but have not yet decided to help contribute to it.

What it doesn't do is it doesn't make anybody buy your product and it doesn't really solve the cooperation problem. I'll explain. So, if we go back 20 years and we look at what was happening in Linux development, it was necessary to have a legal instrument like GPL that you could hang over someone's head and say, “You really ought to contribute back to this, or else you're going to have legal problems,” and everybody was very scared of that and didn't understand it, and so forth.

Then if you fast-forward, particularly to after the internet bust was over, everybody was cooperating on Linux all of a sudden. Well, why did they do that? Did they do that because of GPL? Not really. They did it because they figured out that it was better for everybody to cooperate on infrastructure software, because then they could innovate around other things instead of having everybody having to repave the roads on their own.

I look at all of the licensing changes as milestones along that path. How do you get from a point where people aren't contributing to a he point where they are contributing? Often, you need at least short-term. You need a legal instrument to do that, and that's what all these are more or less. All of these business models and licensing models are living things and they are likely to change over time, and I already said, I think we’re constantly in in a state of change.

So what these licenses do is say, “No, you can't just take our stuff and not contribute back to the maintenance and development of it, because you're starving us.” That is probably the best they can do. They always are instrument serving a business model and a technology model, not the other way around.

[00:47:18] JM: All right. Well, let's move on to business models in practice. Again, Karthik, you're the CTO of YugaByte, which is a distributed SQL database, and you are actually open source I believe by the most traditional open source standards. So how does being open source affect the go-to-market strategy for a database company, or does it even have an effect on going to market?

[00:47:49] KR: It does. It definitely does in some sense, but it's not necessarily what I'm going to say with respect to how it affects us is not anything new. It's nothing that's earth-shattering or something that people would not have realized. The reality is that if you rewind more than 10 years ago, when anybody was shipping a database piece of software, people would take it and use it for free. Like Matt said, until they would hit a level of criticality, they would want to pay. If they hit that level of criticality, they would pay anyway.

But, obviously, between free and between mission-critical, there is a gray area. You can call it semi-mission-critical, somewhat critical, maybe critical, all of these kind of like gray workloads. The licensing and keeping enterprise features in the close is a great way of monetizing those set of users. Again, not every one of these semi-mission-critical or whatever type of workloads would pay, but there's a good chance a bunch of them would. So that yields to a bigger revenue and so on so forth.

But a lot of that has changed today with the advent of the cloud and cloud becoming mainstream, right? So it is pretty clear like in the case of Atlas, in the case of Aurora, in a lot of cases, that people are willing to pay to operationalize databases. The amount of digital footprint has exploded and is actually still exploding overtime.

A lot of businesses or every business out there is becoming a digital business and they're becoming technology companies, which means they need to store their data online. These companies would rather have the database vendors of the cloud providers or somebody who is more equipped to run databases, run databases for them, rather than figuring out how to run it by themselves. So it's not that they want to take the database infrastructure software and operationalize it themselves and run the whole thing and make sure it works. They just didn't have a choice, and now they do.

So the first step to the change we made is the acceptance that the future is bright for database as a service and not so much for shipped enterprise software. Whether it's a private database as a service and a private cloud or public or a hybrid, that doesn't matter, those are just semantics.

So, in the sense that that realization is there, that started making the decisions cleaner, clearer and simpler. So what we are all focused on is that the database has a lot of value that people should feel free to run by themselves. As they start getting into how do I manage this database to make sure it continues to run the way I expect for my mission-critical workloads, we start focusing on a database management platform.

Now, we build software for the private cloud so that people can run their own database as a service platform and we manage, we just recently announced our managed version of the database as a service where we will go to AWS or GCP or whatever cloud you want and create the machines and give everything as a pushbutton service and you could just point your apps to.

So, there are a couple of models. This is our go-to-market as far as the product. So it definitely has an implication on what type of features you build and you put there where. But we've made the conscious decision of keeping the database extremely open. Once again, based on learnings from, say, something like Postgres, which is the database that is having the last laugh. It's in fact the rise in popularity of Postgres is even eclipsing that of Mongo. The funny thing is Postgres does not have a dominant vendor behind it, like it's just open source.

So you think that focused efforts of a company would be the unfocused efforts of the community, but that's not proving to be the case. Postgres is a meteoric rise, but Postgres is not built for the cloud. The opportunity we see is rebuilding or building a Postgres that’s cloud native and scalable and works in Kubernetes, in public clouds and so on. At this point, we felt it's best to go for what Postgres does the same way Postgres does it.

[00:51:33] JM: Just a hypothetical, if AWS were to spin up their own version of a hosted YugaByte, how would that affect your business?

[00:51:44] KR: Yeah, that's a great question. So I will first answer the question as if it were going to happen, and then I'll give you my actual analysis of what I think would happen. If AWS were to host YugaByte, it’s just another provider we have to compete with. If there is a really important mission-critical workload that’s being run by somebody and they have started to depend on YugaByte for it, would that person be like more comfortable coming to the authors and the creators of the software to fix bugs, to guide the roadmap, to give them the next five features or somebody who takes the software that somebody else created and runs it for them?

So that's a big distinction, and that is a distinction in actually the mission criticality of the software, which means the more mission-critical the use case, the more they are likely to come to the makers of the software. This principle is what is actually playing out. I mean, it’s not me that made this up. It’s actually happening with even though Amazon and Mongo and – Sorry. Azure have announced competing Mongo APIs. Mongo still continues to do perfectly fine as a business. It's the same with Elastic and it’s probably going to be the same with Confluent. So that's the one-liner on that.

There is actually other value in addition to just being the creators of the software that we can bring to the table, which is multi-cloud deployments, which are becoming almost mainstream in a bunch of enterprises we talk to, and hybrid deployments, where people trying to get from onpremises to the cloud have to run in a startled mode for a very long period of time. In some cases, they think that that'll be their permanent stake. At this point of time, that is the thinking. So there's a lot of ability to innovate.

Now, I'll come to the second portion of my analysis, and this is just being brutally honest, is that with a database like Aurora, if I were sitting in Amazon's shoes and just putting my software engineer hat on, I'd build the features into Aurora and seamlessly drip new features into my customer base rather than taking a completely new offering that offers the same API in a slightly different variant, then that'll cause more confusion.

I mean, people as it is, complain about so many AWS services that it's very hard to put something together for people. I think they'd probably don't want to worsen that problem. So it's just my take on the second part. But your first question, nevertheless, is valid independent of the assessment and it's still defensible.

[00:54:02] JM: All right. Well, let's begin to wrap-up. I just want to get each of your visions for how the world of open source evolves over the next five years. So let's start with you, Matt.

[00:54:17] MA: I am in the middle of the post right now. It hit me the other day that some of the things that we've long wanted in open source but didn't have, like for example I remember years ago, Jim, maybe 2007, Jim Whitehurst saying, "Hey. You know what? For open source to truly be sustainable and for us to get the most out of it, we need enterprises to start contributing."

Well, we're still not there, but in 2019, we have Capital One, Uber, Lyft and some of the new school enterprises that actively contribute to open source. We have some of the old school, like Capital One and a range of others like Home Depot shows up with their open source folks at OSCON. We're starting to see companies as they take software seriously, as they take that software is eating the world seriously. They realized that that means that they have to take open source seriously, because that's how great software increasingly gets built. So, that's happening.

The other thing that's happening that makes me really excited looking forward is we've talked a lot about the license mechanisms put in place to defend against public clouds taking somebody's software, free writing as Heather said, tongue-in-cheek since that is in fact the whole point of open source, is to enable some degree of free writing.

But we have the public clouds that are increasingly contributing fantastic software. Both contributing to existing projects, as well as contributing their own, so you have like Microsoft Azure Functions. The whole basis for their serverless products going forward, that’s open source. Amazon AWS has Firecracker. They said this is how we operate internally with these little micro-VMs. They can like us. Then Google has been, of course, may be better than others, or at least faster to market than others with TensorFlow, with Kubernetes.

So we have these public clouds that are actively starting to contribute. We’re still having the same conversation about, “Hey, but they’re taking away my business.” But if we look forward to what they're doing, what they’re increasingly doing, it's an exciting future where they’re giving away the keys to the kingdom of how they operate and are saying to may be more pedestrian enterprises, “You can do this too.” We’re seeing things like containers rocketing in part because of the contributions that first Google, now, Microsoft and AWS are doing. I think we’re going to see that continue and we’re going to see it accelerate.

Again, I think we’re entering the Golden age of open source where we have enterprises contributing, and the biggest threat to open source as in some people's minds is actually arguably the biggest opportunity for open source because of the scale and the quality of the code that these companies are contributing.

[00:57:19] JM: All right. Well, let's wrap up with Heather. What are your thoughts on the future?

[00:57:23] HM: Well, one thing I should mention is that one of the categories of licenses that we've been talking about today, the one that I said didn't exactly have a universal name, the source available licenses, and I have recently started a project called the Polyform Project to draft some standard ones, and I thought that was really important because one of the sort of hidden benefits of open source licensing, just from a licensing compliance point of view and analysis point of view, is the fact that they are largely standardized. There's a long tail of open source licenses, but almost all code is under about six of them. That's been actually an enormous boon to the technology industry, because software licensing is complicated and figuring it out is complicated. So I'm trying to make some of the same inroads for the source available licenses that we've had for open source licenses in the past.

But the other thing I would say is that it's easy to get distracted on sort of political arguments about how, for instance, if you write a source available license that’s going to kill open source. Nothing will kill open source.

The reason that open source is alive and well and will continue to be alive and well, is that it's a model that has proven itself out. It took a fairly sort of radical licensing model at the time to convince people that they were better off cooperating than just competing and hiding what they were doing from a technology point of view. But I'll put my hand up as the old person, one of the old people in this discussion too, but I have to say that the contrast between what we saw 20 years ago and what we see now in terms of the cooperation among big businesses with a lot at stake, it's staggering.

I mean, the cooperative model has proven itself, and that will never go away and is bigger than licenses, and it's bigger than discussions about whether SSBL is truly open source and all of the sort of inside baseball. The development model and cooperation has won, and nothing will take it away.

[00:59:51] JM: Well, just to draft on that as a closing note, this is something I felt at each of the KubCons I’ve gone to, and it's something that makes me really like going to KubCon even as KubCon has really grown quite large and I guess it's got kind of a “corporate feel” now. What is beautiful about it is that it – What it represents is the reaching across the aisle in collaboration with each other and forming communal bonds, forming positive-sum relationships with each other because there is an acknowledgment in the software community these days that, “Okay. The pie is big enough. Let's just start working together. Let's just be friends.”

Because if we don't – I mean, there is a competition, of course, but there's – Like you said, like cooperation is winning, essentially. It just makes sense. So it's refreshing.

Thank each of you for coming on the show. It's been really fun talking to you. This topic is inspiring to me and it's very fun to cover. I remain a little bit confused by the licensing discussion. So, Heather, hopefully we'll have future discussions where you can explain these things to me in a little bit more legal detail.

[01:01:11] HM: Always happy to do that.

[01:01:13] JM: My mom is a lawyer. So, actually, I grew up with an affinity for legally. So I'll be pleased to go further in that direction. Karthik and Matt, thank you for blessing me with your company as well.

[01:01:28] MA: Yeah, thanks for having us.

[01:01:29] KR: Thanks for having us, Jeff. Yeah, it was great.

[01:01:30] HM: Thank you very much.

[END]