**Guillermo:** [00:00:00]

Everything on vercel scales with how much traffic you get. It works if you have one visit a day or it works for the largest websites on the internet, we're not that concerned about the CPU or the database. What makes the world go round is the thing you've added. On top of that, the business logic, the data transformation pipelines, the front end and the user experience. So I think what we're gonna what's going to happen is that right now we're in a very fast moving time where lots of models are being updated, but at some point that's going to slow down. And I think we're going to enter the era that we have today with the cloud, where of course, we're all using all kinds of models, but we're going to go and be more concerned with what are we offering to the end user. We have to give them a great user experience, first of all.

**Craig:** [00:00:46]

I might be the most important new computer technology ever. It's storming every industry and literally billions of dollars are being invested. So buckle up. The problem is that AI needs a lot of speed and process power. So how do you compete without cost spiraling out of control? It's time to upgrade to the next generation of the cloud. Oracle Cloud Infrastructure, or OCI. Oci is a single platform for your infrastructure, database, application development and AI needs. Oci has 4 to 8 times the bandwidth of other clouds, offers one consistent price instead of variable regional pricing. And of course, nobody does data better than Oracle. So now you can train your AI models at twice the speed and less than half the cost of other clouds. If you want to do more and spend less, like Uber eight by eight and Databricks Mosaic, take a free test drive of OCI at Oracle.com slash Ienai. That's oracle.com/i on a I. Hi I'm Craig Smith and this is Eye on eye. In today's conversation I had the pleasure of speaking with Guillermo Roche, CEO and founder of Vercel and the creator of Next.js. We discussed the seamless integration of generative AI applications on the Vercel platform, and how it's powering the next wave of dynamic and personalized web experiences, from the intricacies of deploying AI-driven applications to the democratization of AI technology. Guillermo shed light on the evolving landscape where developers are increasingly leveraging AI in their workflows. As we navigate the discussion from the technicalities of AI models to the practicalities of their applications and various industries, it's clear that Vercel is at the forefront of this transformative era.

**Guillermo:** [00:03:01]

My name is Guillermo Rauch, and I'm the CEO and founder of Vercel. Uh, I also created Next.js, which is one of the most popular front-end frameworks in the world. It powers a lot of the interfaces of generative AI companies like ChatGPT. Any time you go to a web browser, whether on mobile or desktop, and you type in a URL, increasingly a lot of those sites are being served by Next.js and in Vercel, especially when they're very dynamic and personalized.

**Craig:** [00:03:34]

Yeah. And actually Next.js, um, uh, I was reading that you, you were into JavaScript when you were very young, which, you know, says says a lot about you. Is there not that many people, uh, that even, uh, as adults, uh, get excited about JavaScript?

**Guillermo:** [00:03:55]

Yeah. It is funny when I was very young, I started doing all kinds of businesses online. And I got really interested in open source technologies and communities around Linux and other programming languages. But one of the things that really stood out to me was when I started doing jobs online for other people, I joined this platform called Script Labs, which is a freelancing platform. I noticed that I could like, solve problems with lots of different programming languages. But one programming language, JavaScript, stood out because it was the only programming language that could run in a web browser, and at the time, it didn't think of it as like, it's like some like 4D chess and super smart MBA, but it's almost like an unfair advantage, right? We have 7 billion people. Almost everybody has a web browser in their pocket or their laptop computers, the most dominant software platform in the world. And this programming language is the only one that can run in those web browsers on the client side, like meaning as close as possible to the interaction with the user. So at the time I didn't this wasn't so obvious, but it did seem to me like JavaScript had this special significant bit.

**Guillermo:** [00:05:19]

And I became an expert in that language, which powers basically every front end experience in the world today. Mhm. If I was everything from ChatGPT to Amazon.com to when you type on Google and it produces suggestions, right. Rich interactive dynamic experiences that are always been built with JavaScript. So the way to think about Vercel, our company is that we took that language and we made it massively accessible to every company on the planet. We built frameworks around it so that you don't have to reinvent the wheel every time you start a new project, and we focus a lot on giving businesses a great ROI. Meaning you put something online and it will make it faster. We'll make it more engaging, more personalized so that people return. So we do very well with e-commerce websites. But now increasingly, everybody seems to be building AI products on our platform. And that's where I kind of like dove in into this world of AI. And it seems to be like all I think about and talk about.

**Craig:** [00:06:23]

Yeah, yeah. So what's the relationship between Next.js and Vercel? And then can you describe I've seen you describe Vercel as a service. Other people have described it as a platform. But but can you talk about that? Yeah.

**Guillermo:** [00:06:39]

Yeah. The way to think about it is you program it with Next.js. You design and conceive your front end experience that a website and that web application. But then you have to deploy it so that you can give people a URL to access it. Right? So you have to ship it. You have to make it go live. Vercel is the easiest, fastest and most scalable platform to ship those Next.js applications to the world. And we support other frameworks that are similar to Next.js as well and have been inspired from Next.js. So in a nutshell, we're really simplifying the problem of shipping something to the cloud. You and I, we met at Reinvent, uh, in in Vegas. Anybody that tries to use the cloud today, it almost seems like you need a PhD. You need to learn like all the different layers of AWS and Google Cloud and Azure. But that's not how developers want to ship applications. And now increasingly, they're building those applications with AI. And the AI really knows how to write your code and so on. So the idea of burdening the developer with provisioning infrastructure in 2024 is not only very costly, but it also slows businesses down. Let's say that you want to ship an AI initiative for your business. You could take the path of I'll build my own infrastructure, I'll configure AWS and do all of that. Or you can ship on Vercel and you go from months to weeks to days to minutes to make something live.

**Craig:** [00:08:18]

And so versus Vercel runs on AWS. Right? So it's an alternative to, you know, all of the AWS tools, SageMaker, bedrock, all of that stuff.

**Guillermo:** [00:08:31]

Correct. It's basically an alternative to figure it all out in this massive marketplace of options. Right. When you go to Vercel, we offer you templates. So for example, one thing has been super successful is we give you the interface of ChatGPT. We give you the front end of it. We we build a clone that's heavily inspired by ChatGPT. Obviously it's not exactly ChatGPT, but folks can now deploy their own ChatGPT. In two clicks. So compare that to the experience of like booting up EC2 instances and triggering virtual private networks and private relays and Nat and all that. Linux, right. It almost feels like, you know, I've been spending 20 years learning this stuff. I don't want anybody to have to learn all those, like, deep details, uh, to, in order to ship an AI application. So going back to that ChatGPT experience, we open sources kind of as an idea. We had an idea about a year ago. We said, like, okay, ChatGPT is cool. A lot of you are going to use it, but enterprises are going to want their own. They're going to want their own models. They're going to want their own retrieval to their own data sets. But most importantly, because of Vercel is so concerned with the front end experience, the brand experience, we also hypothesized people are going to want to embed that experience into their websites and applications and make it their own.

**Guillermo:** [00:10:02]

Their own user interface components, their own style, the, uh, maybe their own sort of representations of the data so that you're not just rendering text. You could render buttons, you could render galleries, you can render maps, you can render charts. And that bet really paid off, because now we have a lot of customers that have launched. The way to think about it is it's almost like a domain specific ChatGPT that you can create. There is a company called Finn Tool that built the Bloomberg terminal of I. You and I were talking about it last time and you were interested because it sounds so cool. Uh, there's another one called Finn Chat, also on Vercel. Think of it as like talking to your stocks. Right. So I think that idea of using natural language to interface with applications is here to stay. And it's not just going to be ChatGPT everybody is going to want this technology. And now Vercel is basically democratizing it for everybody to launch and deploy and customize in a couple of clicks. Yeah.

**Craig:** [00:11:03]

And one thing I've found interesting is, um, bedrock, uh, you know, gives you a it's also you, you can deploy across different models or, or use different models, but it doesn't, um, include GPT four chat, GPT, uh vercel. And when you say, uh, build on top and uh, chat, that's one of the options. But you can build on, on any number correct models. Right.

**Guillermo:** [00:11:34]

So the way to think about Versalles architecture, the architecture we brought to market is that we're separating the front end, which is the user experience from the back end. Typically software was built very monolithically. I talked to a lot of our customers. For example, Chico's moved from a Oracle monolith that they had investigated for 20 years to a composable architecture where the part of the website that interacts with the customer is separated and connected over the network to the different services that supply it. Think of it as like the when you sit down at a restaurant, you're being served and the atmosphere is all awesome. And then the back end is the kitchen, right? So we let you plug in whatever back end you want into that front end. So when you use our ChatGPT template, for example, we let you connect to GPT four. We let you connect to llama will let you connect to Mistral. And customers, I think from a business Defensibility perspective, are very interested in this, right? Because instead of coupling all the pieces, you're basically stuck with that choice for a very, very long time. Now they're finding that Vercel is sort of giving them a lot of freedom. No. And this is true not just for AI applications. It's also true for e-commerce, where your storefront is separated from the content provider, the order tracking system, the search engine. So we're going into this world that's highly modularized.

**Craig:** [00:13:11]

And, uh, something that came up, actually, I met with Rubrik, who you introduced me to, uh, yesterday, and we were talking about, uh, using different foundation models to power something that I'm working on. And I was asking, uh, is it possible to blend models to, to get outputs from different models and consolidate or maybe do a, some sort of a reconciliation of one model has something that.

**Guillermo:** [00:13:47]

Yeah, absolutely. In fact, in fact, this is already how a lot of the AI services at scale work. So it's kind of like the dirty secret of AI is that sometimes you think you're interacting with one model because the industry right now is so concerned. It's so funny because there's so many of the days of like Pre-cloud or the early days of cloud. The industry is very concerned with models and hardware. And that's a strange, because when you think about modern cloud and modern applications, we don't know what hardware is running it. And the analogy for the model would be like, we're not that concerned about the CPU or the database. What makes the world go round is the thing you've added on top of that. The business logic, the data transformation pipelines, the front end and the user experience. So I think what we're going to what's going to happen is that right now we're in a very fast moving time where lots of models are being updated and replaced and so on, but at some point that's going to slow down. And I think we're going to enter the era that we have today with the cloud, where of course, we're all using all kinds of models, but we're going to go and be more concerned with what are we offering to the end user. We have to give them a great user experience first of all. So going back to your specific question about combining models, and I was mentioning when you use ChatGPT and other AI products, they're already intelligently routing your query through an ensemble of models. Some things might be best answered by a smaller model. Some type of query might go to another model where also the UI changes when that model responds.

**Guillermo:** [00:15:41]

The best example is like Dall-E, which generates images. So you enter. Please draw me a photo of a cat. It's almost like you're going through a different pipeline, right? I think that's awesome because at the end of the day, it's about the user experience. I don't care if it's one model, ten models, 100 models. I've heard a lot of our customers are also finding, okay, I'm starting with a very powerful model that can reason extremely well, but then once I notice that there's clusters of queries that the users make. I can first send them to a cheaper model. I just heard from a customer today who is actually doing first summarization with a smaller model that is really good at summarization. Retrieval and summarization, and then this summary is handed on to the very powerful reasoning model. So the very powerful reasoning model deals with the smaller context window and fewer tokens. So there's an analogy in human beings, right? If every support request for Vercel got routed to the CEO, it'd be very expensive use of our time, right? I could probably answer a lot of technical questions, etc. but over time we try to like route them to this specific agent, right? Human agent that can best answer that question. Yeah. In in many cases, by the way, they would be better than me. Because they're so trained on answering a kind of support request that I might use the wrong language. I might forget a reference to the dogs and things like that. So that's what we're seeing with this idea of the ensemble of models.

**Craig:** [00:17:25]

What about I mean, I understand that about routing different parts of a query to different models that are appropriate or relevant for different tasks. But I've been looking across models for, uh, historical summaries, for example, and I'm trying to get beyond find one that that has the least amount of hallucination. Uh, and if I ask for models the same question, I get, uh, four different outputs. Maybe three of them agree. And the fourth one has something that doesn't appear in the other three. I don't know if that's a hallucination, but it seems it if if there were an overlay that then could rationalize those four into a single answer, and you take out the outliers just because probably, yeah.

**Guillermo:** [00:18:23]

You're, you're touching on you're touching on a couple of really interesting points. Number one, the biggest uphill battle that I as a field has today is reliability. Reliability on two fronts. One is just pure service reliability. The infrastructure is so immature that right now is very expensive to run models, and the quality of service is not there. So for example, Vercel gives you a very precise, slow service level objective on latency and availability. And if any of that gets violated, like it's a huge anomaly, right? It's like stop the world kind of thing. When I use AI services, I'm noticing all the time there's degradations. So the field is still pretty immature, right? But then there is reliability on the quality of the answer. Maybe you run the same question three times and two are good answers and the other one is not so good. So that is giving. That's basically giving space for a new field or sort of category of applications and services to emerge. Then deal with the monitoring problem. How do you monitor an LM such that it's always giving high quality answers? How do you benchmark the things that you change about the model when you fine tune it, when you change the prompt, when you replace the model? How do you gain confidence as a as an application developer, as a business owner that when you modify the prompt.

**Guillermo:** [00:20:05]

Or release this new version. Things got better. And what does better even mean? It's a really difficult question. So there's tools already in the marketplace. There's a company, for example, called Brain Trust that the way to think about it is, and I use this analogy, a lot of like software 1.0 and software 2.0, software 1.0 was we grew up in highly deterministic classic programming languages, Java C plus plus PHP. In that world, one plus one is always two steps in the software 2.0 world, where we're not writing the code ourselves, we train a model with data and then we infer. One plus one is two most of the time. Which is a weird thing to say because we're now building all this. We're building the future of this thing, right? So how do you kind of, like, wrap your head around in your arms around that instability of the beast? So in software 1.0, we use testing methodologies. We use to control and verify the software. We use unit testing, integration testing, end to end testing, all kinds of testing methodologies.

**Guillermo:** [00:21:24]

In the software 2.0 world. We're now inventing the analogous services of that. So for example, brain trust, the way to think about it is like unit testing for your LM apps. It allows you to write kind of. And again, this is our highly non-deterministic tasks. But you evaluate that the AI is mostly on track over time, and then you instrument it as well with user feedback. That's why in every AI app, you notice that there is like a thumbs up and thumbs down and there's some signal that's coming from the user. So that goes back into that system. And to your question okay, so if three models are saying one thing and another model is saying another thing, you're going to use platforms like this in order to ascertain which is the right model for you and your task. And how can you prompt it or fine tune it so that over time you have no, let's call it recalls. Right? Uh, and folks are not saying, wait, that's wrong. Uh, or you churn a customer, the customer canceled their subscription, saying like, ah, this tells me this is wrong all the time. How do you use copilot, by the way? The technology that autocompletes your code.

**Craig:** [00:22:46]

I'm sorry. Say again. So.

**Guillermo:** [00:22:48]

Copilot by Microsoft. Copilot by Microsoft. Yeah. Intum. So copilot has an interesting signal in it, right? Because it suggests what code you're going to complete it with, and then you take it or not take it. So that's like an implicit button of thumbs up, thumbs down. So what the people that are working on those products, including Vercel, do, is we monitor the acceptance rate. Other suggestion, and that's why you're going to find that the best AI products are the ones that are suggested. The ones that have actually made it in, in scaled and produce a lot of revenue are suggested because you know that you're sometimes wrong. But the cost of being wrong or is not very high because the user just ignores it. So when I recommend to our engineers what are the best AI products to work today to work on today? They're the ones that put the human in the loop and they act as the. I write her first draft. A new act as the editor and you say yes or no, and maybe you tweak it a little bit and then you ship it.

**Craig:** [00:24:08]

Yeah. I mean, that's a form of reinforcement learning with human feedback, right? Uh, because there's also that, that feedback is then training or fine tuning the model. I've talked to a lot of people about reinforcement learning with AI feedback, which to me sounds a lot more efficient, but also can work in real time. I mean, this human feedback, if I'm if I have an application and I'm serving customers, uh, an output in real time, I can ask them. I can't give them a choice and ask them which one they think is best. I need to serve them an output. And if I have someone. A human in the loop is the obviously it's not going to operate in real time. Um, have you looked at this AI feedback? A mechanism that I know that, uh, MIT is doing some research on it. And I and.

**Guillermo:** [00:25:09]

Yeah, I think this this is a very active area of research because we still have not figured out what is the best way to scale the reinforcement learning phase of training the models. And to your point, I think one thing is really important to call out is when you're iterating on an application. Your only tool is not just retraining the model. What you were mentioning earlier is that today is very costly and slow. Taking the model, doing more RF, that takes a long time, but there's other tools that developers have in their toolbox. They can modify the prompt, they can augment it with other data. They can chain different models together. So I see it as a critical product metric that not only informs model development, but it also informs the developer. How do you even present the data to the user? How do you decorate the prompt and embellish the prompt and add more data to the system over time? But you always have to have that metric and then work backwards to how you improve quality over time and reliability.

**Craig:** [00:26:22]

Uh, the, uh uh, I mean, what some of the what you mentioned is, is referring to Rag. Right. Retrieval, augmented generation. Correct. Um, to me, that that sounds like a short terme solution that eventually models will will be again through RAAF or or, uh. Yes.

**Guillermo:** [00:26:48]

I would love to think so. Like like I would love to call it temporary, but right now, at this point in time, it almost seems like the opposite is true. Because one of the biggest enemies of AI is clearly is that reliability problem we talked about, right? Hallucinations, etc. and outdated data. Not just that. Right. So I'll give you a very concrete example. One of the really cool things that you can build with Vercel is, as I mentioned, like we give you all these templates to add AI capabilities into your products. One of our solutions is how do you add search? To like documentation, websites, content, websites so that you can ask an AI to solve a problem for you. And this is really cool because you could imagine that the customer is going to go to ChatGPT. But you don't control ChatGPT. So you don't know if it's up to date with your data, your documentation, your content, your blog posts, etc. and you also get loose the feedback loop now, like because that's happening somewhere else. You don't know if people are getting the answers to their questions, and if the if the answer is a hallucination or not. So we advocate for here's all the infrastructure and we make it really, really easy to like build yourself out of your product. And one of the things that we found is you almost should completely distrust the knowledge contained in the in the raw model that comes from is training data set.

**Guillermo:** [00:28:28]

Because the way to think about the model is that you're making the model memorize the stuff. And in the process of memorizing or asking you to compress. Forget certain things, and from that process you get reasoning. So that means that the objective of training the model was reasoning, not memory. If the objective was memory, it would be a database. And if the training data set was one terabyte, the model would be one terabyte or maybe less after gzip and some compression. Right? Which means that really what you're getting is a reasoning engine. The data of the world is far larger, and the production of that data far outpaces the training process. In fact, the average enterprise probably generates the volume of data that was used for training some of these models per day, per week, per month. So that relationship continues to hold, that you're producing lots and lots and lots of data. And then you want AI systems that can work with it. Because customers will always have real time questions, right? We or I already lived through this when it first came to the Valley, MapReduce was all the rage, big data, etc. and lots and lots of offline asynchronous processes.

**Guillermo:** [00:30:06]

In order to get an answer to a question, you had to wait two weeks. And then snowflake happened. And now everything is real time. We ingest data to the minute, and then we ask with SQL, that's the big data solution to that. We're going to face similar pressure with AI training. A model is the MapReduce job that takes frickin months, weeks, whatever. But then we're going to have lots and lots of real time data and questions. And I think rag from that point of view is probably here to stay for a while. I think in this space, nothing lasts more than like a couple quarters right now. But it's a fairly stable solution right now that, you know, uh, I have a concrete like end user example of. I use perplexity. I write because perplexity has real time answers, and I've kind of trained myself like my muscle memory right now is if I'm asking if I'm working on a reasoning problem like, um, writing something and summarizing it, or spin or rewriting something, I go to ChatGPT. And I have that back and forth with the model. But if I want real time knowledge, my muscle memory is now perplexity all the time. And the magic there. I mean, there's probably many pieces to it, but of course, they're doing real time retrieval, uh, in order to augment.

**Craig:** [00:31:41]

They're not retrieving from a vector database, which a lot of Rag is doing. They're retrieving through search. Right.

**Guillermo:** [00:31:50]

I think it's probably a combination. I don't know the details, but. The way that I would summarize the transition away from Google into this new systems is the transition from a. Keyword reasoning and keyword targeting. To. It's almost like a spatial reasoning if you think about it in terms of vectors, right? You're approaching this space of the concept. And. The magic of perplexity is that I don't have to know the exact keyword. I can talk to it almost. When I talk to a friend, I'm like, remember when we watched that thing? I think it was a couple years ago. We're in San Francisco and like, like you can talk to the eyes in that kind of like, you know, approximated way. And that to me seems like the durable change because that's more normal and more how we think and how we interact with the world, whereas it always feels like. Google is truly a tool with a learning curve, like you had to think about it in terms of like, I know how to Google, which is still true today. Like, I'm a pretty good Googler, but I don't know if that's a durable, uh, skill. Like, I probably need to take it off my LinkedIn.

**Craig:** [00:33:16]

Yeah. Uh, so, so at the very beginning, you were talking about, uh, writing an XJS and then deploying on Vercel, but Vercel, you can write, uh, in, in any language. It's not, uh, doesn't require nextjs, does it?

**Guillermo:** [00:33:32]

It doesn't require Next.js we have support for. Over 35 frameworks that are custom tailored to writing that user interface side of things, right? Our focus is still on. For example, I just met a customer who wants to build the fastest checkout experience on the internet. That's our joint mission. That's why they're partnering with Vercel. And they're choosing one of these frameworks because that's how you do it today. One of the things that we're trying to disabuse ourselves from as a global community of developers is that every single project requires building your own tools. That's the old way of using the cloud. Like you sit down, you always start from scratch, from an empty canvas. I've met countless enterprises and developers who in the past were building their own frameworks prior to starting projects, and the way to think about it is I used to joke, the problem with building your own tools is not just all them sunk cost is that your developers will not have StackOverflow when something goes wrong. The downside now is you build all of those tools. And now the I don't know how to solve the problems. And now nowadays developers are working with AI, whether you like it or not. They have AI in their chatbots, they have AI in their editor. And now Vercel launched our own AI product called B0, which of course, in the interest of creating user interfaces, what B0 does is you type in a text description of your user interface or even a screenshot, even a napkin sketch, and we convert it into the code of a user interface. So nowadays developers are expecting that I will output really high-quality code that works with this frameworks. And guess what? As soon as ChatGPT came out in New Nexus. Why? Because there's so much data on the internet about Next.js. So I would, I would call it almost like we're at the end of that phase of software engineering where every new project meant that especially the larger companies had to recreate and rebuild their own tools.

**Craig:** [00:35:56]

I might be the most important new computer technology ever. It's storming every industry and literally billions of dollars are being invested. So buckle up. The problem is that AI needs a lot of speed and processing power. So how do you compete without cost spiraling out of control, it's time to upgrade to the next generation of the cloud. Oracle Cloud Infrastructure, or OCI. Oci is a single platform for your infrastructure, database, application development and AI needs. Oci has 4 to 8 times the bandwidth of other clouds, offers one consistent price instead of variable regional pricing. And of course, nobody does data better than Oracle. So now you can train your AI models at twice the speed and less than half the cost of other clouds. If you want to do more and spend less, like Uber eight by eight and Databricks Mosaic, take a free test drive of OCI at Oracle.com slash Eye on AI. That's oracle.com/Eye on A I mean this. So if someone builds on Vercel this space is moving so fast. And just this question about how long Rag will be necessary. I mean I can see the if you're a a company with, with a lot of proprietary data. Um, yeah. It makes sense to, to use Rag, a vector database. But if you're doing something more general and you don't have your own data, rag seems to me a very. A difficult way to to go about it, because you have to go out and gather data to put into a vector database and that sort of thing. Uh, how how flexible? Is virtual for enterprises as this they build an application and then. You know, using rag, for example. And then rag falls out of favor and there's some new way. Yeah.

**Guillermo:** [00:38:11]

Great question, great question. I mentioned earlier that one of the architectural shifts that we brought to the industry is that separation between the front-end code and the back-end. And I agree with you 1,000,000%. Every piece of your back end, you have to think of it as throw away. Everything is modular. What do you need to be concerned with is the user experience. Is are you serving the customer? Fast, error-free, personalized and Vercel makes it really, really easy to plug in these modules. In fact, by the time this goes live, we will have brought a one click integration to Vercel in partnership with a leader in. I don't want to spoil it in case it goes out a little earlier, then later in rag so that you can have a better search index for your application with one click. Okay in. This is not a one off instance where you have to think about how much memory do I have? It scales serverlessly. Everything on virtual scales with how much traffic you get. It works if you have one visit a day, or it works for the largest websites on the internet. And all of the storage connectivity modules that we've built, like we have Postgres, we have Redis, and now we're going to have vector search. All of that also scales Serverlessly. So you can keep throwing data and data and data. Now to your point. Let's say that you say look, my vector index is just not as good as I thought it was going to be. And now this MLM provider is saying, well, I don't need it anymore because they are going to come and crawl my data and I want to replace that. Now, you don't have to throw away your front end code. In fact, you can replace that part of the backend and the user might not even notice. Maybe you just optimize your costs or you improve your latency. So your question is that probably you're pointing at is very aligned with what Vercel wants to solve, which is to bring more durability to to the things that people create. Yeah.

**Craig:** [00:40:30]

And you're you're seeing a lot of stuff being built on Vercel. I mean, I can't imagine you can track everything that's being done, but are there some trends that you see in applications? You know, for me as a journalist, I get pitched all the time for various things, and I can see that, uh, you know, for example, generative AI, uh, text and photo editing tools is just the that market. I it's amazing. But I feel sorry for the guys in that market because, uh, there's there's going to be a, a big come to Jesus moment where the market, you know, the market isn't big enough to support so many tools, but can you see trends that way, and if so, what kinds of trends are you seeing?

**Guillermo:** [00:41:21]

Yeah, I agree on broad strokes that this space right now is intensely exciting, right? There's so many companies being created. It feels very much like when I first came to the Valley, it was like 2008, 2009, like, like history doesn't repeat, but it does rhyme. We're like coming off a huge financial crisis. But at the same time, we had all this new platforms, the iPhone, the cloud. It was the best time to build. Anything now is the same. Where we have a lot of stuff going on. We have wars. We had like financial, like soft landing, hard landing, whatever you want to call it. But we have LMS and vector databases and all these incredible technologies stable diffusion, latent diffusion, etc.. So Vercel supports a lot of these players. So we have as a customer Soonl AI, they're creating music with AI. It's this basically the Spotify of music. It's fascinating. Leonardo, which is the creative platform to ship everything from game assets to illustrations, PCA runway, which are doing like video AI perplexity is a customer of Vercel. We also see a lot of success with like small businesses. There's folks that are perhaps a team of 1 or 2 people, and there's this awesome company called Chat Base, uh, which is growing extremely fast and is like literally, I think that one could be one engineer.

**Guillermo:** [00:43:03]

So there's a lot of demand from consumers, which is a very healthy thing to see. When I talk to these entrepreneurs that are telling me, you know, it used to be that we lived through a lull, you could call it zero interest rates, where folks would raise money, and then maybe they'll figure out revenue. When I talk to this I companies is revenue. And then we'll figure out maybe to your point longevity the visibility you know. But there is there is customer interest. And that's a great place to be from a market perspective. Uh, so the Harvey I is a great example of like legal tech as well. So I mentioned thin chat and thin tool. What seems to be happening, which is really exciting, is there's for every standing category, there's going to be an AI native player. For example, there is Adobe, which is not an AI native player. They're rushing to add AI features and kudos to them. I think they're doing a pretty decent job. But then there's going to be the AI first Adobe iMovie Leonardo. There's another platform called create AI.

**Guillermo:** [00:44:12]

In this. Folks are taking a different path. They're not rebuilding Photoshop from scratch only so that then they can add the Magic Eye tool. They're only building the magic I tool. So what gets me excited is that. A lot of the AA native players might become disruptors because of how little they have to build and how different would they build is. Whereas now the other side of the aisle, like the incumbents do, have a ton of benefits. Think about slack for a second. Slack has so much data about our workspace. When they added the AI features, they have the data. And this is why, by the way, Rag is going to be necessary. They're not going to be able to today train a custom model every second, every time a message gets added. But when they gave me that feature to ask the slack workspace with I. It's going to get some usage right. Because it's there and because we're already working there. So I wouldn't underestimate the I. Improvements or tailwinds that the incumbents are getting. I think Salesforce will get some significant tailwinds. But again, the beautiful thing about AI is that it's going to disrupt what a lot of these applications look like, period. Yeah.

**Craig:** [00:45:41]Uh, are there any industrial verticals in particular that you think, uh, you're seeing a lot of activity in? I mean, you mentioned a couple of financial, uh, start ups. Yeah.

**Guillermo:** [00:45:53]

There's a lot of interest in health care. I think it's going to be a big trend in 2024. Um, there's a company called Luma Labs, uh, that just launched a they call it a genie. They can do generative 3D. It's absolutely incredible. I think generative 3D, uh, spline is another company is adding AI capabilities. Generative 3D is really, really exciting because a lot of those industrial processes, um, involve designing and simulating so many of these processes in three dimensions. Right. In. Until now, it seemed like that was beyond the realm of AI. Everything was about generating images, and now we're starting to hear about generating 3D meshes. And that's extremely exciting to me because one of the questions is if you want to create a video game, if you want to create a some kind of like character animation, this is an open question in my mind. Are you going to be using. Basically an output of a matrix of pixels approach, like the video models like or like you're just generating frames. And only in the internal world model of the LM does it have a conception of 3D. Or are you going to use a tool that actually outputs the 3D models? So it's unclear. And there is awesome company sort of now trying both approaches. But I'm excited about that prospect of like, imagine just having an entire Pixar style film or Grand Theft Auto. But it's all AI generated. Yeah.

**Craig:** [00:47:55]

Um, yeah, it's, uh. It's exciting. Um, yeah. You mentioned world models. Um, and that's something I've been interested in. On the research side, I, uh, wave AI is the only one I've seen that's that's close to launching a product off a world model. I don't know, am I wrong? But, uh, but does, uh, Vercel support that kind of, uh, application? I mean, that's, you know, beyond, uh. Lms, but, uh. Using world models.

**Guillermo:** [00:48:31]

One of the things that I strongly believe is that, as I mentioned earlier, the model will be more of an implementation detail of API services that developers can compose together. The best metaphor that I can give you from, like cloud 1.0 is we used to grab software. And throw it into a virtual machine. In operated and moderate and upgraded cells, and that's still done. But what happens over time is that services emerge. That said, look, why do you care about that at the end? And you have to run many when the data set scales and you have to worry about availability zones and multi-region and encryption and backups. Let us run that piece of software that you were interested in and it will expose you. It will give you an API endpoint to hit when you need to add or request data. So this is how lots of successful cloud native companies have evolved. Mongodb works like this, right? I used to be a self-hosted for a small web. In nowadays I can go with one click, say mango. Please give me serverless MongoDB Atlas. And now I've lost like is there a database? Is there a hard disk somewhere? Is there an SSD? I just use the product. The same thing is going to happen to I. Where when you call that GPT four endpoint, there might be ten, 20, 30 models on there. Would you just care about that input and output? In from a virtual point of view. We love that because we're all about facilitating that rapid iteration velocity for developers, and we don't want to burden the developers with having to think about all this overhead of like provisioning GPUs and batching inference and, you know, error rates and sort of retries and buffering and queues and all of the things that make inference at scale really reliable. Instead, just call the service. In that word for cloud 1.0. My guess is that it's going to also work for the AI generation of cloud native applications.

**Craig:** [00:50:54]

Yeah. Uh, okay. Well, uh, yeah, we're we're, uh, almost to an hour. Uh. So I'll go with that. Well, let me ask one last question. How how is Vercel growing? Because I can imagine, uh, with your your focus on, on serving I native companies that it's, uh, it's expanding really quickly.

**Guillermo:** [00:51:23]

Yeah. It's been, uh. Yeah, it's been amazing. So to give you some context, um, just in 2023, Nextjs was downloading 225 million times. Uh, we have a million monthly active nextjs developers. Um, we serve trillions of requests annually on the Vercel platform. We're in 20 global regions, so we serve traffic from all over the world because our obsession is really with performance and personalization and serving the the most relevant experience to you wherever you are. So yeah, it's uh, and the other, uh, great point of pride for us is that when we look at the lists of the largest generative AI companies in the world. A seven out of the top ten are using Nextjs, and most of those are using Vercel. So we were very happy to ride this wave. And we also fundamentally believe in the technology and its potential to make a better world. So, um, excited to keep investing.

**Craig:** [00:52:34]

And I said that was my last question, but one other popped in my head. You were talking about serving different models, uh, kind of, uh, in the kitchen. Uh, so, so the front end doesn't have to know about it. Uh, or not sure. You know, switching between do you have a is there does Vercel itself have a dynamic orchestration layer that identifies which models are, uh, best for which tasks?

**Guillermo:** [00:53:05]

Great question, great question. Not at the moment. Uh, that requires today the developer swapping things. I actually recently posted an example. Uh, I built a demo app on Vercel, which is Hacker News generated by AI. So I wanted to kind of dog food our tools a little bit during the holiday break. And one of the things that kind of blew my mind was so I initially used GPT 3.5 to generate the stories. So for those that don't know, Hacker News is basically like Nerd Central, and there's all these stories about engineering, etc. And I got some pretty good results. So I got some good speed from OpenAI and some good results. But I wanted to try Mistral, which is the OpenAI of Europe, and they recently came out with this open model that is a mixture of experts. And it's it's a very high quality model. And I was curious, okay, what would it take to I had already written OpenAI code in my sort of the back end part of the app. Right. And I asked myself, what would it take to replace with Mistral? And the answer was. Two lines of code. So the OpenAI client API is becoming almost like the standard. It became like the ad hoc standard. And now all of these other platforms, like any scale compute together I. All this model providers have compatibility with that API. So in this case, and of course this is a very small app, I really validated that thesis of like how easy it is to replace some of the backend pieces. Right. And. This is what also is motivating Marcel to sort of like, let's make it really easy to shop for AI models. That's why we're creating templates integrations. Let's make it competitive. Also for the model, providers like developers should get the lowest price possible per token. So it's going to be an interesting 2024 from a model competitive landscape, right?

**Craig:** [00:55:19]

Um, I've been talking to a company. I think they're still in stealth, but they're building this sort of orchestration, a tool that dynamically routes queries to to different models depending on, on various, you know, cost or latency or accuracy or something like that. Yeah. Very cool. Um, okay. Well, let's let's leave it there. Um, I want to ask you very quickly. You know, I met with the rubric guys. We're very excited about the project. I don't expect you to remember it, but, uh. It's going to cost me. Uh. 30 or 40. Pretty penny and and I need angel investors because I'm a journalist. I'm not a rich guy. Uh, and uh, Sarum said, hey, talk to Guillermo. So when it gets to the point, can I pitch you on that?

**Guillermo:** [00:56:17]

Yeah. I mean, I'm I'm always I'm always interested in hearing out, uh, ideas and whatnot. Um, I knew going in like, this is a very, very high level team, right? Like, they, they're working at a very cutting edge of technology. They work very closely with the Lange chain team. They they ship also, uh, the decision between a lot of consultants and these guys, like, they actually can ship. But yeah, if maybe, maybe if I come across somebody that's cheaper, I don't know, I'll think about it. But, uh, keep me posted. I'd love to see if I can.

**Craig:** [00:56:49]

And you mentioned.

**Guillermo:** [00:56:51]

I do have the.

**Craig:** [00:56:52]

Music app and your Hacker News app are those. Can I find those somewhere?

**Guillermo:** [00:56:57]

Yeah. Uh, I posted on the, um, I posted on my Twitter about, um, the clone. It's it's also open-source, by the way. So you can check out the code and, um.

**Craig:** [00:57:11]

Yeah, I'll check the Twitter if it's on and then.

**Guillermo:** [00:57:13]

Sue Noah and then sue. No. Why is the um, okay. Spotify okay. Great.

**Craig:** [00:57:22]

Great. Well, it's been fascinating. Awesome.

**Guillermo:** [00:57:24]

Yeah. Great chat. Yeah. Super enjoyable. Yeah. Thank you so much.

**Craig:** [00:57:28]

Hey, I might be the most important new computer technology ever. It's storming every industry and literally billions of dollars are being invested. So buckle up. The problem is that AI needs a lot of speed and processing power. So how do you compete without cost spiraling out of control, it's time to upgrade to the next generation of the cloud. Oracle Cloud Infrastructure, or OCI. Oci is a single platform for your infrastructure, database, application development and AI needs. Oci has 4 to 8 times the bandwidth of other clouds, offers one consistent price instead of variable regional pricing. And of course, nobody does data better than Oracle. So now you can train your AI models at twice the speed and less than half the cost of other clouds. If you want to do more and spend less, like Uber, eight by eight and Databricks Mosaic, take a free test drive of OCI at Oracle.com slash Ienai that's oracle.com/Eye on AI. I hope you find the conversation as insightful as I did. That's it for today's episode. I want to thank Guillermo for his time. If you want to read a transcript of today's conversation, you can find one on our website Eye on AI. That's E Y E hyphen o n dot AI. And remember, the singularity may not be near, but AI is about to change your world, so pay attention.