EPISODE 364

[INTRODUCTION]

[0:00:00.0] JM: The history of computing can be thought of as a series of ideas rather than objects. From Aristotle's formalization of the syllogism, to Alan Turing's model for an all-purpose computing machine, to Satoshi Nakamoto's distributed transaction ledger. These breakthroughs did not come in the form of polished tangible objects, like an iPhone. In fact, the objects which end up changing computing fundamentally are often built up from a collection of ideas that seem trivial at first glance.

Chris Dixon is a general partner at venture capital firm; Andreessen Horowitz, and he's the author of the article *How Aristotle Created The Computer*. One job of a venture capitalist is to be early in identifying the ideas that will evolve into influential tangible objects.

In this article, Chris examines several instances in the history of computing where ideas that looked weird and impractical at first glance ended up being world changing. Some recent examples that we discussed are blockchains and neural networks. I really enjoyed this episode with Chris, and I hope you do too.

[SPONSOR MESSAGE]

[0:01:22.0] JM: For more than 30 years, DNS has been one of the fundamental protocols of the internet. Yet, despite its accepted importance, it has never quite gotten the due that it deserves. Today's dynamic applications, hybrid clouds and volatile internet, demand that you rethink the strategic value and importance of your DNS choices.

Oracle Dyn provides DNS that is as dynamic and intelligent as your applications. Dyn DNS gets your users to the right cloud service, the right CDN, or the right datacenter using intelligent response to steer traffic based on business policies as well as real time internet conditions, like the security and the performance of the network path.

Dyn maps all internet pathways every 24 seconds via more than 500 million traceroutes. This is the equivalent of seven light years of distance, or 1.7 billion times around the circumference of the earth. With over 10 years of experience supporting the likes of Netflix, Twitter, Zappos, Etsy, and Salesforce, Dyn can scale to meet the demand of the largest web applications.

Get started with a free 30-day trial for your application by going to dyn.com/sedaily. After the free trial, Dyn's developer plans start at just $7 a month for world-class DNS. Rethink DNS, go to dyn.com/sedaily to learn more and get your free trial of Dyn DNS.

[INTERVIEW]

[0:03:20.6] JM: Chris Dixon is a general partner at Andreessen Horowitz. Chris, welcomes to Software Engineering Daily.

[0:03:25.0] CD: Thanks for having me.

[0:03:25.7] JM: You wrote an article called *How Aristotle Created The Computer*, and this was something that really intrigued me. I want to start by talking about it. You argue that the history of computing can be thought of as a history of ideas rather than a history of objects. If we were the history of computing through objects, we might talk about the transistor, or the mouse, or the iPhone. If we're telling a history of computing through ideas rather than objects, what are those ideas that we should be focusing on?

[0:03:57.2] CD: Yeah. By the way, I didn't mean in the essay to take away from, obviously, the importance of things like the mouse and the transistor and the abacus and the Babbage machine. These things are all very important. I just had felt that a lot of the histories had sort of only focused on those aspects of it as supposed to what I think is sort of another important thread which is how a lot of computing core concepts of computer science came from mathematical logic and, before that, philosophy. The point of my essay is to try to kind of explain how that kind of the genealogy of those ideas and how some of these kind of key ideas around computer science came from mathematical logic and philosophy dating back to Aristotle. I kind of do that through the lens of two seminal papers from the 1930s; Alan Turing, his paper on computability and Claude Shannon's analysis of relay circuits which are widely considered to, I

guess, sort of the canon of computer science, and also happen to come right at this transition period when these ideas are kind of going from theoretical to practical, because, of course, World War II is when a lot of the kind of first real computers were built and then you have the transistor and other kinds of things and then the rest was a sort of meteoric rise of computers and computer science.

These two papers in particular I see as kind of transition, the key kind of transition period and have the nice feature of very clearly sort of spelling out where their own ideas confirm. I try to kind of, in the essay, go through that sort of — I don't know — The history or genealogy or however you want to describe it of how those ideas kind of developed.

[0:05:43.2] JM: It's pretty simple to draw a line from the insights of Claude Shannon or Alan Turing to where we are today and see their impact. It's less clear how you draw that line from Aristotle, but when you look at it from the point of view of Claude Shannon or Alan Turing standing on the shoulders of Aristotle's, it's a little easier to intuit.

Aristotle started with the syllogism, which is A leads to B, B leads to C. That means A leads to C. He phrases all men are mortal, Socrates is a man. Therefore, Aristotle is a mortal. Why does the idea of the syllogism lead to eventually the iPhone?

[0:06:26.2] CD: First, on the method — Just one thing I'll point out is, and actually, I had an original version of the essay where I do much longer excerpts from the text, the primary sources themselves. My claim that like, for example, Claude Shannon comes from Aristotle. It's directly in Claude Shannon's paper. He has two — Rather, he has — Directly in his paper, he has basically two footnotes. One of them is a logic textbook and the other is George Boole who wrote a book called *Laws of Thought* and about 80 years prior to Claude Shannon. Literally, those are the only two references in the paper and the whole paper is about how, basically, Shannon's insight, that you can take Boolean logic — This sounds obvious to us today as people that who know software engineering, but it wasn't obvious in the time. His insight was you can take Boolean logic and map it on to electrical circuits, right?

Claude Shannon actually says he had that insight because he took a philosophy class as an undergrad at University of Michigan and took this logic class and this sort of esoteric thing, this

guy George Boole, who was trying to figure out — By the way, then you go read George Boole, which I excerpt some in the essay and I went back and rad all these stuff. If you just go back and read the preface of George Boole's *Laws of Thought* which is what Claude Shannon is talking about, the whole thing is about Aristotle. He literally said, "This book is an extension of Aristotle." It's not ambiguous.

I sort of just want to emphasize that, like this is not — If you go read these texts, it's actually a very interesting exercise. I used to do this a lot, which is go read historical texts. Read the prefaces, read who they cite, and then compare that and sort of like think of the software engineering point of view. You're like drawing a list of pointers, like this class inherits this class, or something. If you go do that with the history, like it's often very — I find it's often very different than what the textbooks tell you.

Anyways, Claude Shannon very clearly is saying, "Look, I'm taking what Boole did and I'm showing the kind of mapping on to electrical circuits." If you go read Boole, his book is called the *Laws of Thought,* and what he says in, he says, "Look, Aristotle had this idea, which was that there were underlying rules to how people think in the same way that Newton had the insight, that it's not just a coincidence that the way an apple falls from a tree has a similar mathematical properties to way the moon orbits the earth or something."

Obviously, what Newton did was he found these underlying mathematical principles, and what Aristotle was trying to do with his logic was to find these kind of underlying principles of how people think. Specifically, Aristotle's insight was that if you just look at the logical words, so logical words being words like and, or, therefore, not. If you just take the logical words, you can judge the validity of an argument just based on the logical words. It doesn't matter when you make an argument like; Socrates is a man, all men are mortal. Therefore, Socrates is mortal. You can replace Socrates with some other person and the argument is still valid, right?

Then what Boole did was Boole said — Boole came after the kind of Cartesian revolution of mathematics, which was kind of what Descartes did was take really kind of introduce sort of modern — At least to the west, I should day. Introduced modern kind of algebraic notation to mathematics, and Boole said, "Hey, why don't we take this kind of algebraic notation and extent what Aristotle did and go all the way instead of saying Socrates is a man, etc., like put in

variables and put in mathematical operations and then let's see once we do that, we're kind of, now, untethered by natural language and we can go and use all these great machinery that mathematicians have built over the hundreds of years, the prior hundreds of years, and really kind of explore how logic works." That was what Boole was doing, and he very explicitly says that he saw his project as an extension of Aristotle.

I think it's interesting too because — And stop me if I'm going too long. I don't know. I could talk about this topic for a long time. The other interesting thing, I think if you read a lot of modern textbooks. Obviously, everyone knows Boole, and there's Boolean operators in computer science and in every programming language and things. You read about them today and people talking about like a mathematician, or a mathematical logician or something.

If you go back and you look at the historical context, he was kind of a whacky person who — There is no field of mathematical logic. There's literally nothing. There had been no work in logic, period, almost — There's a few of like people in these kind of religious figures in the 13th century or something. For the most part, there had been no work in logic since Aristotle — It was not a field. Immanuel Kant, the great philosopher said, "Logic began and ended with Aristotle." It was considered to a finished topic.

Boole was kind of this — Kind of crank, and I say this in a loving way, like the genius, but at the time was sort of considered as strange person who think you can mix algebraic formulations with Aristotle and do this, and why would you even want to do this is another big question people ask.

What happened is it kicked off a whole new kind of project which is also something I write about in the essay, which is this whole kind of golden period of mathematical logic which really began with Boole and led up through people like Gottlob Frege, who's sort of under-appreciated, kind of massively important philosopher-logician. Then Bertrand Russell who's more widely known who co-authored a book called the *Principia Mathematica,* which was kind of the real attempt to really build all of mathematics out of logic. Kurt Godel incompletes theorem is kind is famously known, who that was sort of kind of maybe the end of the golden period, you want to say like 1860 to 1930ish or something when his incompletes theorem came out. There was this sort of period where all these guys were — And I can talk more about their project while they're

working on it. Their actual goal was nothing to do with computers, nothing to do with any practical. Frankly, they were philosopher kind of logicians, but they basically built all of these machinery which then Shannon and Turing, through their insights, basically were able to say, “Hey, wait a second. All these machinery you built, we can just kind of shift it over from philosophy to computer science, electrical engineering,” and four years later you have a working computer.

These kind of papers and these insights they had were kind of, to me, the critical turning point where they took this very kind of theoretical thing that had happened and had this really important realization and that it could be applied practically, and that really really accelerated the development of computers and computer science.

[0:13:03.3] JM: As I was reading the article, I found myself wondering why is Chris Dixon spending his time analyzing the history of computer science and tracing it back to logic? There’s so many things you could think about in the present, today, that are exciting and groundbreaking in terrifying/world-impacting/even just intriguing. Why are you going into the past?

Reading between the lines that's I drew, and you could tell me if I’m wrong, was if you look back at this history, you see a number of instances where people had ideas that were initially regarded as trivialities or just like totally useless. It's interesting, the further you look back at a triviality that ended up having an impact, the more you see that it just compounded into something really really really important.

[0:14:00.3] CD: You said it very very well. I think — Actually, the original title of the essay, and I was actually thinking of writing a series of essays, it was going to be something — I had this idea, like Nothing Interesting Is a Waste of Time, and the kind of concept was exactly what you said, which is if you go back and you look at so many things which ended up having significant impact on the world. So many kind of intellectual ideas.

Many, many, many of them started off as kind of tinkers, hobbyist, cranks. Again, I’m saying these all in a positive way. Academics, researchers, intellectuals working on what a lot of people thought were just strange esoteric projects.

Yeah, my day job is to try to — I work at a venture capital firm in California and we try to find technology companies to invest in. In some ways, my day job is to try to kind of predict the future to some extent, and I'm very interested in trying to do that. Also, just as a person, my hobby is to read history and I enjoy it.

Yeah, you're right. I think, one; the best way to sort of try to predict the future is to understand the past. It's the only dataset we have to analyze, right? Two; I think there's this very common pattern. I think, for example — I don't think that history is fully have been written yet, but I think when it is, the work going on with neural networks over — Let's call it 19 — Whatever, 40 through 2005 or something, was — Those guys, a lot of them, these like Geoffrey Hinton and [inaudible 0:15:25.1], like these researchers were considered kind of like this weird cultish side group of artificial intelligence, like why are they still working on neural networks when they're not working?

Now, it's like the center of all technology is deep learning, neural networks, etc. That will be another example, I think, when the history is properly written of these people working on these sort of interesting things that seem like trivialities or something else.

Machine learning itself, by the way — I have a friend who is doing machine learning and he's a professor at Penn. He's doing machine learning in the 80s. It was like this weird cultish group. At the time in artificial intelligence, the kind of dominant school of thought was believed you would be building in kind of explicit knowledge. You'd be explicitly teaching computers how to play chess through a set of like heuristics and other kinds of approaches as supposed to having a machine learn. That's seemed crazy. Given the constraints they had around like CPUs and everything else, storage, etc. back then, it was kind of crazy.

Now, machine learning is almost synonymous with AI, so dominant. Computer graphics is another great example. There are great books about University of Utah in the 70s where, literally, everyone from Apple, to Pixar, to Adobe, all those people were working. They're all at University of Utah in 1970s. Every modern computer graphics kind of researcher/company of significant impact was University of Utah in the 70s and that was because some donor gave money to do computer graphics, and it was this weird thing, and these 15 people in the world

who were excited about computer graphics, they all just went to the one place where they could get a job and all of these incredible stuff happened.

To me, it is very common kind of repeating pattern where there's a small group, generally, very very smart people who are motivated kind of through — What I would call kind of pure intellectual passions. Not sort of profit motive or even kind of practical application who are kind of considered weird or trivial or something who end up creating all of these very important things. I think that this is kind of the mathematical logic is that to kind of computer science. In some ways, it's kind of — Of all these examples, maybe the biggest and the most important one because it kind of puts the — Obviously, computer science itself is kind of the big one relative to these other trends. That's exactly right. That's sort of the broader idea.

[SPONSOR MESSAGE]

[0:18:02.0] JM: For years, when I started building a new app, I would use MongoDB. Now, I use MongoDB Atlas. MongoDB Atlas is the easiest way to use MongoDB in the cloud. It's never been easier to hit the ground running. MongoDB Atlas is the only database as a service from the engineers who built MongoDB. The dashboard is simple and intuitive, but it provides all the functionality that you need. The customer service is staffed by people who can respond to your technical questions about Mongo.

With continuous back-up, VPC peering, monitoring, and security features, MongoDB Atlas gives you everything you need from MongoDB in an easy-to-use service. You could forget about needing to patch your Mongo instances and keep it up-to-date, because Atlas automatically updates its version. Check you mongodb.com/sedaily to get started with MongoDB Atlas and get $10 credit for free. Even if you're already running MongoDB in the cloud, Atlas makes migrating your deployment from another cloud service provider trivial with its live import feature.

Get started with a free three-node replica set, no credit card is required. As an inclusive offer for Software Engineering Daily listeners, use code "sedaily" for $10 credit when you're ready to scale up. Go to mongodb.com/sedaily to check it out. Thanks to MongoDB for being a repeat sponsor of Software Engineering Daily. It means a whole lot to us.

[INTERVIEW CONTINUED]

[0:20:02.6] JM: The esoteric ideas today that seem to be getting born out, like you said, one is neural nets and another might be the cryptocurrency community and the — Let's say, there's probably genetic engineering or quantum computing, these things, or even space, where you can say, "Okay, we are at a point where we've made some fundamental breakthroughs and the rest of the work is just incremental progress, and we know how to make incremental progress.

[0:20:34.8] CD: Yup, that's right. I think —

[0:20:36.1] JM: Does that sound accurate to you?

[0:20:37.7] CD: Absolutely. One interesting thing I wrestle with is I feel like the things that I'm arguing here, more people understand how this happens. One thing is that there are these kind of driving forces behind — Just talk here just about the computing industry, right? There's things like Moore's law that just computers have gotten faster. Morre's law, by the way, is often, I think, kind of thought of just as like transistors, just as sort of what they call [inaudible 0:21:06.1] scaling of packing more transistors on the semiconductors. Really, it's more of a broader kind of economic principle that just as the computing industry focuses on certain things, like computing and networking, and storage. They've just gotten a lot better every year or two. Sorry. They doubled roughly every two years or something like this in terms of their performance.

As an example, Skype back in 2002, people had dial-up networks and broadband penetration was — I don't know. What? 10%, 15% or something in the developed world and computers didn't have microphones built in. I don't know. The routers would block peer to peer connections. There's a whole bunch of issues. All that stuff got fixed, and Skype went from being kind of dismissed as a silly toy to fundamentally Skype an things like it replacing landline phones as our primary form of communication.

That's an example where you can just kind of — Or YouTube was originally like little dial-up stuff and small bad quality video and not a great library of content. You could kind of see YouTube and project it forward and imagine how it would become something much larger.

I think smartphones — I don't know. The great example where the first version of the iPhone, right? It was only AT&T and it dropped calls all the time and there were only a couple of apps. We've seen that movie before. We've seen the PC and the Apple computer and all these other kinds of computing platforms grow. Some of it is just sort of like you kind of look at these trends and you see how they play out.

I think the other thing I think a lot about is going back to something like cryptocurrency. I would argue that the Etherum, Bitcoin, kind of broader cryptocurrency community is probably the largest and smartest software engineering organization in the world. There are probably 20,000 programmers, if not more, working on this stuff. There's a lot of really really smart people. I spend a lot of time with them and I'm just constantly impressed. Do you want to bet against those 20,000 really smart engineers who are like super passionate about this and working on it all the time? I wouldn't want to bet against them.

I think the mistake people make is they look at these things and then they look at this technology in a snapshot and they say, "Okay. Well, there's X, Y problems with Bitcoin or Ethereum." I don't know if Ethereum is this kind of newer cryptocurrency, which I'm pretty excited about, I'm very excited about.

[0:23:28.6] JM: Sure. I [inaudible 0:23:28.9]

[0:23:29.9] CD: Okay. Yeah, you can go and you make a big list of issues with Ethereum, but they've got a great development team. They've got a great community. You could go back and imagine critiquing — I'm sure Linux had all sorts of problems over the years, and especially like when it was first created. You got to kind of look at these things as dynamic processes that evolve overtime, and you got to sort of say, "Okay, given the trajectory and given who's working on it, what are the underlying trends, what are the winds at its back? What are the headwinds?" and sort of try to map that out and make a prediction as to where it can go.

[0:24:05.2] JM: Is there some sort of pattern among the personalities of the people working on this? Like Claude Shannon, or Alan Turing, certainly weren't motivated by money, or as far as I know. It almost seems in my interactions with the people in the cryptocurrency community, the people who are most influential don't seem to be concerned with money at all. Maybe part of it

is intellectual curiosity. Frankly, it almost seems like they're more interested in just like, "Hey, let's see what happens when we mess this thing up. Let's just destroy the status quo." They're really interested in poking the status quo and seeing what changes.

[0:24:46.4] CD: Yeah, I think there's definitely an element of the community that's sort of disruptive or something. I think you're right, they're not motivated by money. Look, people have made money in cryptocurrency, if any rational person, let's say, in 2011 and like, "I want to make money." You were a cryptocurrency which is like a completely crazy idea to — Way to do it. You go work on Wall Street or something, right? It just was not a rational strategy.

Now, it turns out in retrospect, that people have made money in cryptocurrency. No, it's very hard to imagine any rational person sitting down and saying, "Okay, I'm going to go make money. How should I do it? I'm going to go find this cryptographically secured distributed ledger and I'm going to buy a bunch of units on it." It just would not append of the candidate for any rational person.

Almost by necessity, like it's — Almost by like definition, the people that have made money on it or people that didn't care about money and they cared about distributed systems and cryptography and all the other kinds of things.

Yeah, Claude Shannon was a pure researcher and as as Alan Turing, and I think they were — I've read history, biographies of them. I haven't obviously didn't peer into their soul or didn't know them. From everything I could tell, they were motivated by pure research, pure kind of intellectual — I think, like a lot of sort of philosopher, mathematicians, computer scientists. They were almost motivated by kind of aesthetics. They were building kind of these beautiful structures.

You think about with Shannon — What's so interesting about Shannon — Also, if you read his later work on information theory, he's just a fresh original thinker. What's so beautiful about the symbolic analysis of relay circuits paper that I talked about. I'd encourage your readership. I think if you just Google it and you find it and just even read the first couple of pages. It's great, because the whole idea, you get it in two paragraphs. It's such a big beautiful idea that you can map Boolean logic on to electrical circuits. He was the first one — Just think of it as like this

way. We take it for granted in software engineering now that there's a separation between the logical and the physical layer. That was what the paper was. He was the first one to make that distinction. He was the first one to realize that.

The other thing, we take it for granted that the whole idea of digital circuits, it's actually a very counterintuitive idea. Not to us today, because we've kind of grown up with it. Think about it originally, you're throwing out — What you're doing when you convert a transistor to a zero or one state, is you're throwing out a ton of information. Nature is continues valued. It's not binary. To throw out all of that information is a very non-engineering thing to do, and there's a lot of waste in a sense in turning an analog system into a digital system.

There's a huge advantage, which is what Shannon saw, which is you can — By the time he write that paper, people had shown that you could build all of mathematics on top of Boolean logic. Literally, from the pieces of — It's actually has been shown now, and we know this in computer science and the end game, you just need one Boolean operation and you can construct the other Boolean operations, you can construct arithmetic, what we call an arithmetic logic unit which is a key part of a CPU, but that was all shown by logicians.

Logicians had done that, like Peano's axioms. Peano is an Italian logician who showed — He kind of basically built ALUs, arithmetic logic units before Shannon's paper, before there were electrical circuits. They built all these machinery up, so they had shown, the [inaudible 0:28:18.0] mathematic, they had shown you could build all of mathematics out of the Boolean operators. What Shannon said was, "Oh my God! If electrical circuit can represent a one and a zero," and he actually has this very simple chart in this paper where he just shows whenever the circuit is open, that represents and, and it's just maps each of the Boolean operations on to different circuits.

His insight was, "Okay, we're going to pay this price." At first, it seems counterintuitive, "We're going to throw out the continuous valued nature of nature and analog systems and replace them with digital systems." In return, we get this — These other people have created this machinery that show that you can model all of mathematics. By the way, if you believe Boole, the book was called *The Laws of Thought,* you can model all of human thought if you believe Aristotle and Boole, like people hadn't done it yet. We may be doing it now, by the way. At the time, no one

had done it. That was a theory, really, in some ways Aristotle had and Boole had was that once you can kind of get the basic logical operations, you can model a human thought, the laws of thought.

The really deep thing Shannon — I think it's one of the deepest insights in the 20th century. I don't know. I think what Shannon did is unbelievable. In some ways, what you can say is that Shannon realized that by converting circuits from analog to digital, you can now start to think about electrical systems that think, really, right? Because once you've got logic, you've got thought. It's taken now — Whatever. It will be 100 years, but I think 100 years from when he wrote that paper in 2036, we probably will have machines that say are thinking, and it's in many ways due to Claude Shannon.

[0:29:56.9] JM: I think some of these ideas are easier to look at in a historical context and say, "Oh, of course that worked that way," or "of course that didn't work." One example I think about is I remember in computer architecture class, the professor said something about, "Yeah, there was a period of time where people were messing around with Ternary systems a lot." You didn't just have zeroes and ones, you had zeroes, ones and twos and it's like, "Maybe that works. At some level it seems like, "Okay. Cool, you can make more opcodes or something. Isn't that great?" and then didn't work

Then here we are with quantum computing where it feels like that's something similar. I don't know where to draw the analogies that are valid, because we're coming from the present and it's harder to evaluate at present time rather relative to the historical context.

[0:30:49.9] CD: Yeah, it's fascinating. There's a whole field. Actually, I did my — I was in a Ph.D. program when I was younger in philosophy, and I dropped out of my Ph.D. program. What I was actually focusing on was it was called non classical logics. There's a whole field of logic where you just sort of change the assumptions.

For example, you mentioned the Turnery system. There are systems where you have values like true, false, and neither as an example. It turns out there's what's called [inaudible 0:31:20.2], which is logic that you add in kind of concepts of like necessarily and possibly as an example. There's logics for ethics. There's logics for quantum computing.

I wouldn't be surprised — By the way, it's very analogues. It wouldn't shock me if it turns out later on that one of these really crazy non-classical logical systems ends up being a really core thing. Maybe quantum computing, maybe AI, maybe something else. To give you another historical analogy, Einsteinian physics, it uses non-Euclidean geometry, where non-Euclidean geometry is much like non-classical logic.

Just to give the brief history of it. Euclid, obviously created the element and wrote the elements, which is the great kind of classical text on geometry. He had five axioms. Mathematicians spent — I don't know. Almost — What? 2,000 years being skeptical of the what's called the parallel postulate, because the other postulates all seemed simple and elegant. This one parallel postulate, two lines, two parallel lines, two parallel lines never cross always seemed suspicious and they always were truth of saying, "Hey, can we derive this postulate from the other four?"

Eventually, in the 19th century people said, "Hey, why don't we just get rid of this postulate and see what happens?" That was when you had these whole kind of mathematical area called non-Euclidean geometry, and so there's all these — It's very hard to kind of picture these things. They just seemed like these curiosities in the same way that non-classical logic seems curiosities today. Then Einstein came along and it turned out it actually was the best way to describe the real world. That's another great example, where this kind of — I don't know.

I'm not trying to actually make a prediction about that, but I think the point you make is a great one and it very well could be that there's something that we consider, some weird logical system today that turns out once you have quantum computers, once you have maybe a lot of these stuff going on in AI now, like they're kind of resurrecting a lot of other interesting ideas from other areas of academia. You just don't know where these things will go. I kind of think of it as like these really smart kind of purist kind of quirky cultish groups, like the logicians and the non-Euclidean geometers and the neural network people prior to 2010, etc. They're just sort of creating all these conceptual machinery.

Then when the right kind of real-world factors kind of comes — Something happens and there's a new breakthrough and there's a new need or there's new what, suddenly — And then

someone comes along and says, “Hey, look at all these stuff and we can bring it over and, “Boom!” make something great.

[SPONSOR MESSAGE]

[0:35:59.0] JM: Spring is a season of growth and change. Have you been thinking you’d be happier at a new job? If you’re dreaming about a new job and have been waiting for the right time to make a move, go to hire.com/sedaily today.

Hired makes finding work enjoyable. Hired uses an algorithmic job-matching tool in combination with a talent advocate who will walk you through the process of finding a better job. Maybe you want more flexible hours, or more money, or remote work. Maybe you work at Zillow, or Squarespace, or Postmates, or some of the other top technology companies that are desperately looking for engineers on Hired. You and your skills are in high demand. You listen to a software engineering podcast in your spare time, so you’re clearly passionate about technology.

Check out hired.com/sedaily to get a special offer for Software Engineering Daily listeners. A $600 signing bonus from Hired when you find that great job that gives you the respect and the salary that you deserve as a talented engineer. I love Hired because it puts you in charge.

Go to hired.com/sedaily, and thanks to Hired for being a continued long-running sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:35:32.0] JM: When it gets that esoteric — Like when you’re talking about something like quantum computing or certain niche topics in the crypto community, is there a certain point where you have to just say, “I don’t understand this. I’m not going to be able to understand it, but I’m going to make a bet on a person.” I guess it’s just multiple factorial —

[0:35:53.6] CD: Are you saying in my job as a venture capitalist?

[0:35:54.6] JM: In your job. Yeah.

[0:35:56.3] CD: Yeah, I know I think in my job for sure. I think so much of it, it ends up being a bet on people. I even look back on both good and bad investments I've made. It's funny. Even the good ones I've made, like I go back and I — I always write a memo as to why I'm making the investment, like an internal memo just to kind of describe it. It's just a couple of page kind of thing, like describing the concept behind it.

Even when I'm right, I'm usually wrong. In the sense of like even when the investment works out, the theory behind it — Just the world is so hard to predict. Usually, the one thing that, usually, when I look back in those memos, if I'm right, I got right, was the people. It seems like virtual reality will be a big thing, and these seem like the people that would do it. Then you try to go and be more specific about how the particular product is going to play out and how the competition is going to play out. Kind of everything else besides the people and the general kind of direction, it's very very hard to predict.

With investors with a company called Coinbase, which is a Bitcoin kind of cryptocurrency company, and it's similar thing. We invested four years ago, and it was just the really, really smart passionate team. They were true believers. They were ball from the beginning. They were high technical. Didn't know where it would go. Knew there was something interesting happening in the movement. They've done really great job and they've been successful.

[0:37:21.0] JM: What's funny about that example is it probably seems like it's even less clear where the company is going to go today. Is it going to be a crypto token holding company? I don't know. Maybe you know more than I do.

[0:37:33.0] CD: No, you're right. Look, I would still say if cryptocurrency is like the internet, we're still in the 1980s. I think it's still very early. They've got a plan and everything else, but it's early. There's going to be a lot more twists and turns. If you told me three years ago that Ethereum would be where it is today, I thought you're crazy, and all the other stuff happening and just the innovation. Just the rate of innovation is so high. Who knows?

[0:38:01.7] JM: I want to talk about crypto tokens a little bit. I know we're most the way through our time.

[0:38:06.3] CD: Yeah, I love it. This is one of my favorite topics.

[0:38:08.5] JM: Yeah. You wrote about it recently, and it was a really clarifying article that I'll put in the show notes. The line that you draw is that crypto tokens are kind of a way to do decentralized state management. The block chain allows you to do decentralized state management, but the crypto tokens are the building blocks for that decentralized state management, or at least more formal, higher level building blocks.

State management, the way it works on the internet today, one way you could look at it is like Facebook, which is like a proprietary way to manage the state of your public profile. Explain what the difference between that example and a decentralized state management system is.

[0:38:57.9] CD: First, let me say, I think — I talked about this in the blog post. I think we're living in a very centralized period. I think it's even more centralized than the Microsoft period of the 90s. What I mean by that is the strength of Facebook, Apple, Amazon, Google, they're stronger than ever. You just look at — If you're a software developer today, look, you're dependent on — If you're building an iOS for android, you're dependent, they take 30%. They could kick you off the platform. God help you if you're building on a social network, like Facebook or Twitter. I had a bunch of friends who tried to do that either years ago. They all got like kicked off or whatever. No rational people are still trying to do that, I hope.

It's just a very — If you're building an infrastructure, like an infrastructure company, Amazon will just take an open-source version and add it to AWS and — It's just a very — We're living in a very centralized era, and I think that's — I personally, I'm in the minority in Silicon Valley on this. I think that's a bad thing for the world, and it's not because of some radical political beliefs of mine. I just think it's bad for innovation, because if you go back and you look at it, Google started in a garage in Menlo Park and Facebook was started in a dorm room. In fact, the internet is open and accessible to any kind of hobbyist and tinker. It has been a wonderful thing for the world and has allowed all sorts of innovation to flourish. I think we need to maintain that, and I think it's very important.

Right now, I think there are virtues to centralization. We get these awesome products, like the iPhone is great, Facebook is great, Instagram is great, but I think it's kind of a sugar high. It's good for now, but long-term is going to hurt us. I'm very interested in things that provide a possible new kind of way to open up decentralization. That's said, I was very involved in — I've been involved in technology now for 15, 20 years or something in the industry, in the startup industry. I was very involved — It's called 2007, there was a moment where social networks had become a thing. It was like Friendster, MySpace, Facebook, but there were also like RSS, friend of a friend. There were these open protocols that were vying for it.

If you go back and you read, and I cite some of these in my blog posts. If you read the blog posts and the kind of news articles at the time, it was a real open question 2007, which was social networking would go. Would it go the way of SMTP? Would it be like email? There's an open kind of protocol that's like how you store your friend's graph and everything else, and you choose which client you want to use. It's like the way we use email and Gmail, etc.

Of course, what happened is it went the other way, and centralization won and Facebook won. Why is that? The obvious answer is Facebook built a better product. Why they built a better product? I'd argue that is because the open side, which I was part of, I was there. Just by the way, this is something that I feel like — I don't know. A topic I have some understanding of, I guess. I was there and I was on sort of open side and I was trying to work on stuff.

The open side was fighting kind of with a one arm tied behind his back. Just to give you an example, you talked about state management. Let's just say you were — In 2007, you wanted to create a Twitter competitor using open-protocols, and so the basic functionality one is a person comes on and they claim a username. I am cdixon, right? How do you do that on the open internet? There's no open database that everyone — There is one. There's DNS. DNS is the only one. There's no sort of shared resource database. There's DNS, and there's DNS.

If you look at all these things, everyone tries to overload DNS once again. Email uses DNS. Websites use DNS. If you go look at a friend of a friend and — I don't know. XML and RSS, and everything else, they were all saying, "Okay, crate a website, and on that website create this XML code that shows your friend list, and everyone is going to put that on their website, and

then these crawler are going to go and aggregate it all and that's how we're going to create the friend graph." But that required all these massive coordination. Each user had to do this. Poor DNS was, once again, being asked to kind of do the work of the only kind of shared state on the internet. It's the only shared database on the internet.

I'm sorry. When I say shared, like open, not controlled by any — Yeah, there's some control over, but something you can sort of — When you have your domain name, when you have, I'm C. Dixon, I have cdixon.org, I own it. It's not Facebook's property."

What cryptocurrency does and blockchains and things like sort of the underlying architecture along with a lot of — I think the kind of blockchain has been over-height a little bit, like there's other aspects that are important about cryptocurrency besides the blockchain. One of the many things that cryptocurrency does is it allows you to provide like a state. If you wanted to create a Twitter, or Facebook today in an open way, it'd be very easy to say, "Okay, I am C. Dizon. I claim that username. Here is my friend graph, and you could store all these stuff on blockchain — There's a whole bunch of different blockchains you could store it on." It would be open in the sense that there's no single kind of for-profit company that can control it and take it away from you, but it has the benefits of centralization that you can create a name and it's simple and it's quick and you could create these like modern. There's no reason you couldn't build a modern client around it and it would feel like software made by Facebook or something in terms of the quality, but then it would have the benefits of being no controlled by Facebook, which I think would also be really useful for the world.

There's been just a lot of really really interesting, I think, kind of core infrastructure work. It's still in the infrastructure phase, like it's not — We're basically building infrastructure. When I say we, the cryptocurrency community right now. At some point, it will move into the kind of end-user application phase, but their infrastructure is not done. I think we still got a couple of more years to that. That's sort of this whole, and there's this other really interesting concept, which is concept of tokens, which is kind of this thing that started with Bitcoin but now has moved beyond it and a lot of it is around the kind of Ethereum ecosystem, which is another really interesting idea.

If you look at these platforms, like any platform in the history of computing. Platform being kind of a network that connects developers and users. It could be an operating system. It could be a social network. It could be whatever. Anytime that users and developers interact with each other though kind of a centralized gateway, that's a platform.

If you look at the history of platforms — Sorry. Can you remind me of where I was going?

[0:45:19.2] JM: Sure. I can just ask you a question. You're talking about the history of platforms, and one way to look at it is there's a vacillation between decentralization and centralization. Some people that decentralized state management stuff is just going to completely disrupt the centralized state management stuff. We're all going to decentralize all the banks. We're going to decentralize Facebook.

Even on the blockchain or on different blockchain, you see places of centralization because you find that there is a usefulness to it. If you look at lightning networks as these little points of centralization for faster transactionality, it's an example of like, "Okay. Actually, maybe you don't want a complete —" Maybe there's a place for a relationship between centralization and decentralization that's complementary rather than disruptive. Is that the future that you think we're headed towards rather than all one way or another?

[0:46:18.5] CD: I wanted to talk about — The other benefit of cryptocurrencies is this idea of tokens. If you kind of go back and you look at the history of platforms, and this is like operating systems and social networks and whatever. There's basically just constant battles that go on between the different participants on the network.

Just to back to Microsoft as an example, they obviously had Windows and they had Office and they had these big battles with the kind of the killer apps in their platforms. For example, Intuit and Quicken, there were all these Microsoft money — They were trying to take out those guys for a long time. They had big epic battles with Netscape.

The way I would describe it is if you actually look at the most kind of vicious battles in the history of — Not vicious, but like whatever — Aggressive battles in the history of technology, they're

often between what kind of economist call complements, which is like products that work together, like an operating system and an application.

I like to say the most vicious battles in technology are not between the hamburger and the hotdog, they're between the hotdog and the hotdog bun, like the two things that go together actually end up having like the most — Anyways. You see that, again, Facebook and Zynga. I don't know if you've followed that whole saga. Twitter and TweetDeck and all the other kind of Twitter clients and things and the battles there. Basically, there's this concept thing in technology where you build a platform, the applications come along and then — If you're a developer on iOS and you built like a cool — Whatever. Calendar app, or voice app, and then Apple built the same thing and kind of knocked you off, like every WWEC, there's like who got killed today, or something, right?

There's like this weird kind of thing where it's like symbiotic but then also adversarial between every platform and their applications built on top of it. You just spend a lot of time — I've written a lot of blog posts about this. You just spend a lot of time kind of watching these kind of platform wars play out.

One of the other beautiful things about cryptocurrency is the idea of tokens. The idea of a token is — Literally, like it's a token that is like a thing you can buy, that you can own, or you can be given. The token has value and can appreciate and is traded on an open market. The way these new cryptocurrency networks work is there are systems for kind of giving out and selling the tokens. The way Bitcoin works is miners who are people that kind of are the service providers who host Bitcoin, so to speak, get paid tokens every day in exchange for what they're doing.

The way that a lot of these new tokens are working is like the development team will hold 10% of the tokens for themselves to fund the development. They'll give some portion of the tokens overtime to miners who provide the services. The applications built on top of it, those people will just buy tokens or be granted tokens or whatever. Then the tokens are freely traded on these token exchanges.

What's beautiful about this model is that it aligns the incentives of all the participants so that all of the participants in the network, the core developers, the third-party developers, the service

providers, a.k.a. miners, the users, they all have — The incentive for all of them is to make the token more valuable instead of fighting each other for whatever profits there are or the way that it's happening on every other platform in history.

I think that's one reason why Bitcoin has surprised people. There's a Tumblr that's like 120 times Bitcoin has been dead, and it's sort of a funny little Tumblr, but it's like —

[0:49:45.0] JM: Bitcoin Obituaries.

[0:49:46.2] CD: Yeah, Bitcion Obituary. It's like mainstream media has had — Literally, every month, there's like an obituary of Bitcoin. Of course, it's now in an all-time high and it will continue to be in it. I think that the thing that they underestimate is, "Yes, it's cryptographic money, and yes," blah-blah-blah. All these arguments against gold standard and kind of they tried out all these kind of tulip bulbs and everything else. What they don't get is the power of an aligned network, having millions of people with the same aligned incentives who are also super active internet users and all these other things, and their programmers. You get a million programmers who are all aligned. It's a very powerful force in the modern world, and I think they underestimate that. I think that's what this sort of token thing is doing now is they're sort of taking that beyond Bitcoin and applying it to other areas.

It's very exciting. If you look at Brendon Eich, the creator of JavaScript, he just did a token that's called the Intention Token. By the way, I don't know any of these will work, and I'm not — I should also say I don't recommend people go buy these tokens. I'm not sort of giving financial advice. I think it's all very speculative. We're a venture capitalist. We take risks and expect to lose money on a lot of our investments. I think it's very exciting what people are doing.

I think like what Brendan is trying to do, for example, is he wants to fix the way that monetization works on the internet. If you think about the stuff going on now where you're on Twitter, you click on a link and you get all these popup ads, and so maybe you're running an adblocker. Meanwhile, journalists aren't getting paid enough. Then — I don't know. The only people it seems to be working for are like Google and Facebook.

It's not working for the journalists. The users aren't that happy. Maybe we took a wrong turn. I think what Brendan is arguing — I don't want to speak for him, but he's arguing, "Maybe we took a wrong turn when we bet on these banner ads and all these other kind of stuff." I kind of think we did. I don't think they're particularly good for users. They aren't particularly good for creators, like writers, and journalist.

By the way, it's a really important thing. We need a way to fund journalism. We need a way to fund creative people. The internet should be the greatest thing that ever happened to those people, because now they can be — There's 3 billion people with smartphones. You can write an article, it's a miracle. We have been walking around with these super computers in our pockets with access to more information than the president of the United States had 20 years ago or something. It's a miracle that's what happened.

It should be that anyone could write something now and 3 billion people can read it instantly. This should be a wonderful thing for people that are creative. It should be a wonderful thing — What you're doing on your podcast for people creating videos, people writing things, people creating music, people creating art. This should be a golden age for those people, and it is in some ways, and their work is able to be exposed more broadly.

On the sort of monetization side, making money, it has not been a good thing for them. I think there's two kind of views to that. One is it's just the nature of the internet and it's just as you're trading — They say, digital dollar, whatever, analog dollar for digital pennies and just — I just don't believe that. I've heard that my whole career. I remember people saying that search would never make money, and Google of course did. I remember people saying social networks never money. I remember people saying video games never make money. There were two video game companies last year who sold for $10 billion. They say micropayments never work, and that's of course how video games all make money now. I just don't believe it.

I think when people say there's something magical about the internet that makes it not make money for category X. I just think we haven't figured it out yet. I believe that. I just spent my entire career being told X will never make money on the internet, and it's just always been wrong and I just don't believe it. I think once you get — I just think once you get the user — It's like micropayments is a great example. Everyone said it would never work. This is what — Like

Clash of Clans, it's like three billion in revenue this year of micropayments. Once you make it, you just need the right user experience, you need the right set of incentives, you need the right platform. Who would have imagined that these little ads in the side of the page on search engines would be — Whatever. 80 billion this year in revenue on this? I would have sounded just insane. Anyone who describe the world we live in would have sounded insane. I realized I sound insane, but I think we are going to enter a period where — I think we're getting to a period where it's going to be a golden age — I think it's incumbent on us as people in the technology community to try to find this future, but I believe there is a future there that we can find if we do the right things which will make the internet a wonderful thing for creative people both in terms of getting exposure and in terms of providing financial support for them. I think this is the most promising movement, and that's one of the many reasons I'm excited about this movement.

[0:54:22.9] JM: That is probably what's motivating most of those 20,000 developers that are working on cryptocurrencies on a regular basis.

[0:54:31.2] CD: By the way, it's a fiery revolutionary exciting movement. You talk to these people and they're like — They're just like fired up and they're like, "We're going to save the world —" Maybe they're right or wrong or whatever, but — The really interesting thing too is it's not based in Silicon Valley. I would say it's more Europe and Asia, New York — A lot of it, like Vitalik — I don't know. Vitalic was — I think he was in Canada and Switzerland. I don't even know where he is in Ethereum. Then Gavin, I think, is in Berlin. Also, all the core team of Ethereum, they're all over the world, but not in Silicon Valley. We don't know where Satoshi is honesty, or who Satoshi is.

Generally, if you just look and you kind of head count all the kind of key people in this movement, very few are in Silicon Valley. It's really this kind of global movement. It's led by technologists. They really believe — I think they're very passionate about kind of ideologically based in this kind if Utopian ideological vision which a lot of people make fun of it. I think it's very real and serious. There's just a rate of stuff that's coming — The idea generation and just the excitement. It's just very high and it's high quality software, like go look at — Vitalic writes, and if you go look at the code. There's a lot of really really good stuff being written.

[0:55:43.7] JM: All right, Chris. Well, thank you so much for coming on Software Engineering Daily. It's been really fun talking to you. I've been wanting to for a while.

[0:55:50.2] CD: Yes, my pleasure. It's been fun.

[END OF INTERVIEW]

[0:55:58.7] JM: Deep learning is at the forefront of evolving computing and promises to dramatically improve how our world works. In order to get us to that bright future, we need new kinds of hardware and new interfaces between this AI hardware and the higher level software. That's why Intel acquired Nervana Systems, a platform for deep learning.

Intel Nervana is hiring engineers to help develop this full stack for AI, from chip design to software frameworks. Go to softwareengineeringdaily.com/intel to apply for a job at Intel Nervana. If you don't know much about Intel Nervana, you can check out the interviews that I've conducted with engineers from Intel Nervana, and those are available at softwareengineering.com/intel as well.

Come build the future of AI and deep learning at Intel Nervana. Go to softwareengineering.com/intel and apply to work at Intel Nervana. Thanks to Intel Nervana for being a sponsor of Software Engineering Daily, and I really enjoyed the interviews I have done with the Intel Nervana stuff. I think you'll enjoy them too.
enjoy it to

[END]