Demetri Kofinas:	00:09	What's up everybody? Welcome to another episode of Hidden Forces with me, Demetri Kofinas. Today, I speak with Leemon Baird, the inventor of Hashgraph, an entirely new, distributed ledger technology with the potential to displace the entire ecosystem of blockchain based applications including cryptocurrencies, smart-contracts, and just about anything else you can imagine. Leemon is a genius. He set the record at Carnegie Melon for the fastest ever PhD achieved in Computer Science, clocking out at only 2 years and 9 months. He has over 20 years of technology experience, co-founding several startups, with two successful exits. He has multiple patents and is published in peer-reviewed journals and conferences in computer security, machine learning, and mathematics. In this episode, we get a first-hand look at what may be the future of the Internet. Hashgraph is a distributer ledger like no other. Able to scale faster, safer, and fairer than its blockchain competitors, the assertions made by its inventor and confirmed by its early success with credit unions, payment providers and processors, are astounding. In this nearly hour-long conversation with the algorithm's creator, we learn how he purports to have solved the problem of scalability with significantly improved throughput, security, and fairness, all while eliminating the need for conducting the type of costly mathematical computations inherent in Bitcoin's Proof of Work, and without relying on any of the slower, less secure, and ultimately unproven solutions that have been proposed by the larger blockchain community. The implications of this technology are tremendous, and you are about to be one of the first to learn all about it.
Demetri Kofinas:	01:23	As always, you can gain access to reading lists put together by me ahead of every episode by visiting the show's website at hiddenforces.io. Lastly, if you are listening to this show on iTunes or Android, make sure to subscribe. If you like the show, write us a review and if you want a sneak peek into the how the show is made, or for special story lines, go through pictures and questions, then like us on Facebook and follow us on Twitter and Instagram at @hiddenforcespod. And now, let's get right to this week's conversation.
Demetri Kofinas:	02:00	Leemon Baird, welcome to Hidden Forces.
Leemon Baird:	02:35	Oh, well thank you for having me.
Demetri Kofinas:	02:37	I'm very excited to speak with you today. So this to me is a really exciting conversation. I will say I was not unaware at the fact that there is a distinction between distributed ledger technology and blockchain. However, I was unaware of

hashgraph and when I reached out to some people that work within the blockchain community, it seemed that no one was really aware. No one that I knew of who works in this area, was aware of this protocol. Either that or they were blissfully unaware, but it seems a really remarkable piece of technology. A remarkable protocol/algorithm that deals with basically reaching consensus and solving many of the problems that people have been attempting to resolve, with respect to this sort of this distributed network and not having to go through a third party to authenticate a transaction or ownership, or whatever.

Demetri Kofinas: 03:27 Let's ... I wanted to say this. So I think early on in bitcoin ... with the development of bitcoin, there was this conflation between bitcoin and blockchain and I think that what's happened since then, there's been this conflation between blockchain and distributed ledger technologies. And what I think is really fascinating about your distributed ledger, this hashgraph that we're gonna get into, is that it addresses some of these issues that the blockchain community has been aware of for a long time, and has been attempting to fix. An example being the hard fork in the bitcoin community recently and the proof of stake with Ethereum. Let's start off a little bit to talk about hashgraph and I wanna double back into distributed ledgers and to blockchain and then kind of start to drill down in different areas. Why don't you flesh out the distinction a little bit between blockchain and distributed ledgers for us and then we can proceed from there?

Leemon Baird: 04:18 Oh, sure. So a distributed ledger is just a group of computers that need to come to an agreement on the order that certain transactions happen, and those transactions will be updating some kind of information, maybe the amount of money in wallets or maybe the results of smart contracts. There's various things you can store, but the whole point of a distributed ledger is that a bunch of computers come to an agreement and we don't have to trust any one computer. If some small group of computers are evil and trying to do something bad, they can't hurt. Now of course, if everyone's bad then that hurts us. In fact, there's a theorem that says, "If more than a third of us are bad, that hurts us." But you wanna have a distributed ledger, where all the computers agree, they come to consensus on the order and no small group of computers can hurt the community as whole, and there's lots of ways of doing that.

Leemon Baird: 05:06 There's actually five different approaches to this. There's the proof of work that you started with, with bitcoin and then people moved on to leader based systems like PBFT and Raft and Paxos, and there's a whole bunch of those, and now they're going into ... they say proof of stake. They're talking about sort of simulated economies where we're kind of gambling money on our votes and hoping that we'll come to consensus in the same way that an economy has come to a consensus, and then there's voting systems that go back decades that are so slow. I'm not aware of anyone ever having put a real voting system out there in the real world and then there's hashgraph, which is a voting system with no votes, and so you get all those strong guarantees that we've had for decades, but it's incredibly efficient 'cause we don't send any votes over the internet. So those are kinda the five different ways of doing it and so hashgraph is a little bit different from the other ones you've seen and it gives you some advantages because it's a very different way of doing it.

Demetri Kofinas: 06:04 Let me ask you something. Some of these like for example, blockchain combines a number of these, correct? It is both economy based and proof of work based, correct?

Leemon Baird: 06:12 So bitcoin is not doing what I was calling economy based, what people call proof of stake based. In bitcoin, you have to have your super computer, a mining rig and you have to have lots of electricity to run your super computer, but if you're doing that, then you're just solving these little math problems and every 10 minutes somebody on the globe actually solves it and then they get a prize. They actually get rewarded with more bitcoins. So that's the way that system works.

Demetri Kofinas: 06:38 Economy based is proof of stake? That's what proof of stake refers to.

Leemon Baird: 06:41 Yeah. When people say proof of stake, they're usually referring to these new economy based systems. Technically, you can combine proof of stake with leaders, you can combine it with hashgraph in a very narrow sense. But usually when people say proof of stake these days, they're referring to a system where for example, you have different people proposing what the next block should be, and everyone in the community actually gambles real money, real cryptocurrency on what they think the next one should be-

Demetri Kofinas: 07:06 Right.

Leemon Baird: 07:06 And anyone who votes with the majority gets more money, and anyone who votes with the minority loses some money. And so the idea is, "Oh, we're all motivated to vote with the majority." We'll all be listening carefully to everyone else and trying to

		vote with the majority and then Adam Smith's invisible hand of the market makes us all vote with the majority and we reach consensus. That's kinda the reasoning behind it. People often just refer to that as proof of stake, although technically proof of stake is just anything that involves weighting by cryptocurrency, and so you can have much simpler systems that are also proof of stake. But these days when people say proof of stake, they're usually referring to that kind of a system. I'm just calling that a simulated economy because you're using these economic forces themselves to try to drive the consensus algorithm. Now of course, every system has some kind of economic motivation. Bitcoin miners do it because they get money out of it. That's an economic motivation, but it's not like we're simulating an economy and using that to drive consensus.
Demetri Kofinas:	08:02	Got it. Got it.
Leemon Baird:	08:03	So that's the distinction I'm making.
Demetri Kofinas:	08:04	Got it, got it. Then I wanna go further into this, with respect to the gossip protocol as well at some point. How about ... Let's hold off on that. I don't even know ... I suppose gossip would be considered a protocol that you're using as part of this hashgraph, correct?
Leemon Baird:	08:17	Mhm (agreeing)
Demetri Kofinas:	08:17	And we'll get into that, but before we do, I wanna stick a little bit on bitcoin specifically and proof of work and how that works, because I think that's very fundamental to understanding the limitations of the ... and not just the limitations, but also the costs associated with proof of work. I think make the case for or provide the sort of ... the context for the problem that you're attempting to solve with the hashgraph algorithm. So can you talk to me a little bit about how proof of work ... and to the audience... How proof of work works in forming consensus within bitcoin and how the blockchain incorporates that?
Leemon Baird:	08:48	Yeah. So the whole point is we need to be able to put these transactions in order. For example, if I were to have coins in my wallet, virtual coins in my virtual wallet, and I tried to spend it at two different stores, that would be bad. I'm basically created money out of thin air and so what we as a whole community, we decide which of my two spends, when I spent it at Alice's store and when I spent it at Bob's store, which one gets to count as coming first. And if we all agree that I spent it at Alice's store first, then we as a community agree that Alice gets to keep the coin and Bob is out of luck, and that way we keep the money

supplies constant, and nobody can double spend and forge money. So it's all about the community agreeing on what order these different transfers between wallets occurred and that's really the whole thing you need for cryptocurrency. If we can all agree on what order it occurred, then we can keep the money supply constant and we can make sure no one cheats.

Leemon Baird:	09:41	So in bitcoin, the way it works is anyone in the world can propose, "Here's the next few transactions I think we ought to put into our official list, our official chain, and here's the order I think they should be in." But if we just let anyone do that anytime they wanted to, there'd be chaos, everybody would be doing it at the same time. So what we say is, "Well to add to the next block on the chain, the next container full of transactions on the chain, you have to solve a math problem, and it's inverting a hash, it's not useful to anyone in the world, but it's really hard to do. In fact, you need a super computer to do it." They're called mining rigs and people build chips specifically for this purpose and so all around the world, you have people working really hard to solve these problems all the time.
Leemon Baird:	10:29	Well not people, computers, expensive computers, using lots of electricity to try to solve this problem and whoever solves it first, which is kind of a random way of picking a person. That person gets to add the next block to the chain, and when they add the next block to the chain, they actually get rewarded with some coins themselves, and that's why they're doing it. That helps pay for the super computer and for all the electricity, and so in bitcoin, we can go up to maybe seven transactions a second. In hashgraph, we have hundreds of thousands a second, because it doesn't use that kind of a method, and there's ways of making bitcoin a little bit faster. But right now, it's seven a second, not hundreds of thousands a second, just seven transactions a second.
Demetri Kofinas:	11:10	So for those in audience who are wondering why would you need to solve these arbitrary mathematical problems and why would you need to expend so much energy, let's talk a little bit about why there's a need to incur this cost, having to do with ... really the making the system robust and secure, that's a big part of it. And then, the amount of time that's required to arrive at consensus and how that's a big part of it. I mean the way I see it-
Leemon Baird:	11:33	Yeah.
Demetri Kofinas:	11:33	Those are the two primary drivers for the proof of work in bitcoin, correct?

Leemon Baird:	11:37	It is and ironically, the reason we're wasting all this money on electricity, is to make the system slower. That sounds crazy, but it's actually what it is. So here's the deal, as long as we could all agree on putting our blocks on one after another, and we all agree on what order the blocks are being added, everything is great. But if people are allowed add blocks too fast, then after we have a chain of a certain length, you might try to add the next block and I might try to add the next block. And we really are fighting to see who gets to add the next block because whoever does so, gets some money from it. But if you and I both add the next block at the same time, then what used to be a nice single chain has just forked, and now it's starting to look like a Y shape. It's looking more like a tree and less like a chain. Well that's a disaster, because I could've spent my money at Alice's store on your block and on Bob's store on my block and the community would never know which came first.
Leemon Baird:	12:30	So the whole point for the security of the system, is that everybody has to agree that when you try to add a block, and I try to add a block, which of us was the real one and which of us doesn't get to count, and the way we do that, is we slow everyone down. If a new block is only being added once every 10 minutes, then it's gonna be very rare that two blocks get added right at the same time, and when two blocks get added right at the same time, everybody votes with their feet. People now try to solve the math problem to add to one of those forks or to the other one and whichever one wins, whichever one grows a little bit faster, everybody keeps adding to that one and eventually it gets so long that we all just kind of ignore the short one and it disappears, and whoever did the blocks on the short one lose out. They spent all that effort and they don't even get any money for it, and so we're all motivated to keep adding to the long one.
Leemon Baird:	13:21	And so that's where the consensus comes from, but of course, it takes a while, and we usually say, "Well you need about six of these added on to one of the blocks ... on one of the chains rather than the other one, in order to say we've all agreed on which one was longer." And so we have to make sure that the community as whole around the world is able to extend one of these two branches by six, before another fork happens.
Demetri Kofinas:	13:45	So just to clarify that point, that refers to the time needed to arrive at consensus, the second point, and that has to do with the fact that there is no broad consensus about who should take the next block. From the beginning, the community's competing for that and so because each individual known on the network is competing for the next block, a tremendous

		amount of energy's expended in order to get to that finish line. Once they get to that finish line, there are a number of blocks that could be there at the same time and if you didn't have the time to actually pick winners and losers, and basically impose order on the chain, you'd not only have one fork, you'd have many forks.
Leemon Baird:	14:22	Exactly.
Demetri Kofinas:	14:22	You'd have infinite number of forks branching out like a delta, and you'd have massive amounts of energy used and the network would go nowhere.
Leemon Baird:	14:29	Exactly. We would never reach consensus. We have to as a community, have time to chop off each fork before the next fork happens. Otherwise, they just explode outwards. You get exponentially many forks and the whole system collapses.
Demetri Kofinas:	14:41	And it wouldn't work. Now talk to me a little bit about the security aspect of this, which is incurring a cost, making it more difficult to conduct attacks like DDoS, for example.
Leemon Baird:	14:51	Sure. So of course, it's hard to DDoS this. DDoS is Distributed Denial of Service, that's where you have a whole bunch of compromised computers on the internet. They could be web cameras, they could be printers and baby monitors and little tiny computers around the world. A bad person can get an entire network of these botnet and then have them flood packets into one computer, so much that that computer can't talk to anybody. It can't really receive anything or send anything useful, and you can shut down one computer. If the botnet's even bigger, you could shut down multiple computers, but you have to have compromised more and more computers to shut down a lot. So the system that we have with bitcoin, you would imagine is pretty robust. If you shut down one of the miners, that just means somebody else is gonna win the contest and there first.
Leemon Baird:	15:36	And so in that respect, we have security from DDoS attacks and we also have the security of knowing that we're not gonna be adding blocks that cause forks faster than we can chop them off. The downside is that the only way we can slow people down, is to make it really expensive to add a block. We have to make it where it takes you a lot of time on a big computer with a lot of electricity to add a block. If it were cheap, everybody would do it all the time. If your cell phone could add a block, then everybody would be adding blocks and then we would never be able to chop it off. It would be a Hydra, it's growing

heads faster than you can chop them off, and so that's why bitcoin is constantly chopping things off, whereas in hashgraph, it's allowed to branch all at once and you never have that problem. You could have a node that's a phone and it's constantly adding branches and we don't care. It all works.

Demetri Kofinas: 16:23 And also ... There's also sort of a complimentary aspect to this, which is I think, works for cryptocurrency, but obviously does not work for many other types of things like smart contracts, which is that you are internalizing the energy consumed in the real world into the value of the currency. It took actual work to mine these currencies and ... "to mine these currencies" and therefore, there is ... I think that adds a certain level of value to bitcoin, that isn't necessary for many of the other things that we're attempting to solve with block chain. So that then creates the problem of scalability. These two things that we just enumerated, the cost in terms of making the network more secure and helping it to arrive at consensus, create problems when it comes to scalability, and one of those things obviously has to do with throughput and speed of the network. Why don't you talk a little bit about the challenges to scale with proof of work and then I wanna use that to transition a little bit into proof of stake, because Ethereum is working on that, and then we can get into hashgraph.

Leemon Baird: 17:24 Yes. So the economics, we have to talk about what's the value of a coin. We also have to say how much does it cost you to do a transaction, and so the value of a coin is just supply and demand. It's how many coins are there and how much do people want those coins, and right now the demand is skyrocketing. You have the price going up and up because demand is skyrocketing and supply is fixed. It's actually growing slowly but it's gonna stop at 21 million and there'll never be more than 21 million bitcoins. So if you wanna know how much a bitcoin is worth, it's 21 million divided into how much everybody really wants to hold bitcoins. So that's where the value of a coin comes from. Now on the other hand, you have what's the cost of a transaction, how much does it cost me to move money from one wallet to another, and that can cost a couple of dollars. Why? Because it is so expensive to add another block to the chain and the only reason we're adding blocks is so we can have more transactions. Now of course, if you could put a trillion transactions into one block, that would help somewhat, but then you'd have some other problems that happen. But that's where the price of the transactions come from.

Demetri Kofinas: 18:28 The use of the currency, yeah.

Leemon Baird:	18:29	That's it and the problem is, we can't make it cheaper to mind, because then people would be mining faster and then we'd have this problem of the Hydra head growing too fast. So we can't make it cheaper to mine, but we're also having difficulty making it faster because that's a problem, because it would grow too fast. The forks would happen too fast, and so what we're kind of stuck with is this fairly slow system that is inherently expensive because we're wasting ... Okay. Estimates I keep reading are different but something like 1/2000th of all the planet's electricity is now going to bitcoin mining or 1/1200th, I keep reading different numbers. But an enormous amount of electricity, these are resources that humanity is currently wasting on solving math problems that don't help anything, and the enormous waste there shows up as a high cost per transaction.
Leemon Baird:	19:22	Now if your transactions are a couple dollars apiece, there's still lots of things you can do that are useful. But if you had a system that didn't have that expense, that you could be the equivalent of a miner without having to have a super computer, if you could do it on your phone, then you could talk about a tiny fraction of a cent as your cost per transaction, which is what we do with hashgraph and that gives you scaling. So if we're talking about scaling to lots of transactions, then we really do need to be able to do many per second and we need to have the cost be very low, and I think we're gonna have problems if we're spending a 2000th of all of mankind's electricity on it. We're gonna have problems getting the cost down to a tiny fraction of a cent each, and so that's the scaling problem that we're dealing with, and why do we care? Well if you just wanted to replace credit cards, we're talking about thousands of transactions a second, maybe 10,000 and bursting, in times of high traffic. So seven a second isn't even close and for other applications, we're gonna have to go well beyond that. So it would actually be useful to do hundreds of thousands.
Demetri Kofinas:	20:25	And there's also of course, which is not ... Sorry to interrupt you. There's also something else, which is not related to what we were just describing, which has to do with ordering transactions. And ordering them essentially in real time, which you don't mind ... It's not a problem I wanna transfer money from here to my family in Latin America or Dubai, but it is a problem if I want to make a functioning exchange, commodities market, or if I wanna follow positions of cars on a ... in the real world in real time. So that's something that I wanna get into in a minute as well when we get into hashgraph, specifically in how it differentiates. I do want tie this off a little bit with PoS, proof of stake and help us understand how it is that proof of stake

attempts to remedy the short comings of the block chain and proof of work, and then when we get to hashgraph, you can sort of walk us through how hashgraph is superior to that system.

Leemon Baird: 21:18 Yeah. And so it's useful to talk about ... And I see we've also just kind of skipped over all leader based systems, which quite a few of the permission networks right now are based on leader based. So there's a lot of different terms here, but what you're calling proof of stake is one way of doing things, proof of work is another way of doing things and leader based is a third way of doing things. Those are three different ways of doing even within a single shard, even before you get to sharding. So I can just explain them briefly if you'd like.

Demetri Kofinas: 21:45 Sure, sure. I just wanna make sure that we have a chance to get into the actual nuts and bolts of how hashgraph operates, but let's go ahead. Yeah, absolutely.

Leemon Baird: 21:51 Okay. A lot of these leader based systems like Paxos and Raft and PBFT and there's some systems that combine a leader with other things. Distributed proof of stake actually has leaders, where you take turns being leaders for two seconds. There's another system where people call it proof of stake, but it actually has a leader as well. In these systems, you ultimately end up ... everybody sends their transactions to a leader, the leader just picks an order and then they send them back out, and there's several problems with that. And the problems are if you have a distributed denial of service attack, the DDoS attack we talked about before on the leader, they shut down the whole network by just shutting down one computer. And you can say well that's okay. We have a mechanism in Paxos, where if the leader gets shut down, after a few seconds, we notice hey, Alice went silent. Maybe Alice shouldn't be the leader anymore and then we all elect Bob to be the leader.

Leemon Baird: 22:43 The problem is that if one of our computers has a virus, it knows who the new leader is and can direct the botnet to attack the new leader, and so the botnet plays follow the leader, and as soon as we switch from Alice to Bob, it attacks Bob. Or you could talk about a system like distributed proof stake, where we have a list of leaders and each of them is a leader for two seconds. Well that doesn't help because if one computer has a virus, it knows who the leader is and it just goes down the line. The botnet can attack each leader when it's that leader's turn, during that whole two second window, and so you can shut down the whole network by shutting down one computer at a time, and that's pretty fragile. Clearly you can shut down the network if you shut down a lot of computers, but if you just shut down one at time, you don't want the network to crash.

Leemon Baird:	23:26	So that's the problem with these leader based systems, including some that have proof of stake in the name. Then there these other systems that have proof of stake in the name, which are the economy based systems that I was talking about, and those are where you're gambling money on what the consensus is. And so the idea is, there's never a leader, it's just that we're all self-interested and trying to gamble our money on what the leader will be ... on what the consensus will be. And in those systems, you don't even have as much of a math proof of security as you did for the leader based systems. Basically, you can have instabilities and could we ever prove that the U.S. stock market will never crash? No. Real world economies are really complicated. Can you prove that an economy based system will never do something bad when some of the players are infected with a virus and are trying to lose money? I don't know. And what if there's a firewall separating us into two groups? Well they each reach a different consensus? I don't know. And what if the firewall does ... starts playing games where it turns on and off, or it lets some packets through and not others? Will that stop consensus? I don't know. We don't have any math proofs about how they can survive these attacks.
Leemon Baird:	24:33	So for that kinda proof of stake, we have all sorts of questions about how secure it is and then there's a third use of the term proof of stake. Just to be confusing and that's just somehow stake is involved, somehow money is involved, and lots of systems can do that kind of proof of stake. With hashgraph, we would use that kind of proof of steak. We just say, "Well we're taking votes." But everybody gets voted according to how much weight they have, gets their vote weighted by how much money they have. So the more money you have, the more weight you have in your votes. That kind of proof of stake is simple and it makes a lot of sense, but it's not actually a consensus algorithm it's just something you put on top of your consensus algorithm.
Demetri Kofinas:	25:08	So when Vitalik Buterin talks about Casper, when we hear about what the Ethereum community's attempting to do with PoS, what do they mean and, which one are they using there and can you give me a concrete example of a way in which you'd be concerned that this would be insecure or a way in which you would get around this and compromise consensus?
Leemon Baird:	25:26	Yeah. So there's lots of things that are being proposed and variance of them being proposed, and typically what you find is that they try to become very fancy and be a combination under

the hood of something that's partially leader based. Maybe there's something called a leader or maybe there's someone who proposes a list of transactions to be in the next block, or maybe you have taking turns being leaders. Often they haves some kind of voting being involved, but it's not pure voting. The voting is voting who will be the leaders or something. And then often they have some of this economy thing, where you have to bond some money and freeze it up. So there's a lot of different systems out there being talked about. Often they'll just say proof of stake when they mean a system that's a hybrid of these different approaches and the problems are the ones that I was just talking about.

Leemon Baird:	26:12	If there's a leader anywhere in the system, then you have to worry about DDoS attacks on that leader, and if there is simulated economies anywhere in that system, then you have to worry about a partition, a firewall breaking us into two different sub networks, that each come to a different consensus, allowing double spending. And you have to worry about virus infected computers that are trying to lose the money they've bonded, causing havoc in the community. And there's even stranger things you can do, where a small number of bad players that only have a little bit of coins, but they have a fast connection can start to trick the rest of the community into losing money, and the rest of the community says, "You know what? I'm not gonna go first. I'll let everybody else go first." But if everybody else is letting everyone else go first, then the whole system grinds to a halt. There are lots of game theory on these things and none of those systems have a math proof that they are asynchronous byzantine, and that's really what we're talking about. If you wanna be safe from all these attacks I just said, you have to be asynchronous byzantine and basically, none of the systems we're talking about have those proofs and I think it's because they aren't actually secure. That's why we've never proved they're secure.
Demetri Kofinas:	27:18	Proof of work is byzantine fault tolerant, right? It's just no asynchronous, correct?
Leemon Baird:	27:23	Many of the things called proof of work are not even ... are not even byzantine in any sense.
Demetri Kofinas:	27:29	Not at all?
Leemon Baird:	27:29	Some of them are.
Demetri Kofinas:	27:31	Well I mean like for example ... I mean like bitcoin, for example. Would that not qualify as byzantine fault tolerant?

Leemon Baird:	27:36	No, it wouldn't.
Demetri Kofinas:	27:37	It would not?
Leemon Baird:	27:38	Isn't that interesting? Yeah. Isn't that funny? You can imagine a simple bitcoin system-
Demetri Kofinas:	27:41	Well let's actually ... Sorry. Before we go into the weeds actually, I ... because ... Why don't we explain to our audience and maybe you could give a-
Leemon Baird:	27:47	Yeah.
Demetri Kofinas:	27:47	Definition of what BFT is and-
Leemon Baird:	27:48	Yeah.
Demetri Kofinas:	27:49	What asynchronous BFT is and how that plays into this conversation?
Leemon Baird:	27:53	Yeah. Yeah. So this is a really important thing and the community as a whole throws out the word byzantine all the time. But often they're not really matching the actual math definitions of it and it sounds very abstract, but it turns out to be kind of critical. So what byzantine fault tolerant means is that when you're trying to figure out what the order of transactions, there comes a moment in time when you know that you have reached consensus. Now remember with bitcoin, every time you get a confirmation, you become a little bit more sure, but you never actually reach a moment where you know. You just say, "Well after six confirmations, I think I'm sure enough to give the customer the thing he just bought. Or maybe he's buying something really big, so I'll wait for 12 confirmations." But you never really reach a moment where you're sure.
Leemon Baird:	28:37	Byzantine fault tolerant means there's a moment when you know for sure you have consensus and you're mathematically guaranteed that everybody else is gonna reach the exact same consensus. There can't be any difference and you will never be wrong and you'll never be able to change your mind. You just know for sure that you have consensus. So you reach consensus, you're never wrong and you know that you've reached consensus. That's byzantine, byzantine fault tolerance. But then there's a question. If you have some math theorem proving you're byzantine fault tolerant, how many assumptions did you have to make? Did you have to say something like, "Well I'm gonna start off by assuming there are no botnets in

		the world." Because if you start a math proof by saying, "I assume there are not botnets in the world." Well then I'm not really sure what your proof means, because I think there are botnets in the world.
Demetri Kofinas:	29:28	Right.
Leemon Baird:	29:29	Yeah. So there's something called asynchronous byzantine fault tolerant and that's basically where you start off by assuming, "I think there are no botnets in the world. I think there's no such thing as a DDoS attack." Okay. Then your theorem is valid, but maybe it's not as useful as you were thinking, because I think the world does have DDoS attacks. In fact, there was a really bad one last fall that shut down a lot of people.
Demetri Kofinas:	29:53	I must plead my ignorance on the distinction here. There are many in my audience unfortunately, who are not engineers, we'll experience the same level of ignorance here. Without sort of belaboring the point because I don't wanna get lost in the weeds, what is the major take away here between these two distinctions and how it relates to bitcoin and how it's gonna relate to hashgraph as we're gonna get into it?
Leemon Baird:	30:14	Hashgraph is asynchronous byzantine. Asynchronous means, yeah it still secure even in a world with botnets, even in a world with malicious firewalls, like if you have a country with a firewall around it that can stop all traffic, or can slow down some traffic by arbitrary amounts. Even in such a world, hashgraph is still mathematically proven to be secure, period.
Demetri Kofinas:	30:34	How is it that a DDoS attack could be perpetrated against the bitcoin network?
Leemon Baird:	30:38	Oh, it can't. It's not byzantine, even at the beginning. It doesn't even ... I mean it's not even byzantine under bad assumptions, it's just no byzantine, period. In bitcoin, there is never a moment in time where you know that you have consensus and you'll never be wrong. That never happens. All that happens is you get more and more confident over time.
Demetri Kofinas:	30:56	I see what you're saying.
Leemon Baird:	30:58	And then it could be even worse. What would happen if-
Demetri Kofinas:	31:00	That's interesting.

Leemon Baird:	31:00	Say a firewall shut down, separated the network into two pieces. Could you end up with a chain with six confirmations on the left side and chain with six confirmations on the right side with different answers? You could.
Demetri Kofinas:	31:12	So give me a scenario in which what was perceived to be consensus would be changed. For example something where a transaction that occurred somewhere down along the blockchain is altered. Is that what you're suggesting?
Leemon Baird:	31:24	No, it's worse than that. So I am evil and I spend my coin in two different stores, Alice's store and Bob's store. That's bad. One of those two shouldn't count and Alice and Bob should wait until they get their six confirmations before they give me the product, so I don't get two different products but only pay once. However, what would happen if lots of the miners were in one country that had a firewall around it and the government of that country turned off the firewall, stopped all from going through or stopped all bitcoin traffic from going through? Then all the miners inside the country could come to a consensus with a chain of blocks and all the miners outside that country would come to a different consensus with a different chain of blocks. And if I had spent Alice's store inside the country and in Bob's store outside of the country, Alice might get six confirmations and say, "Yep, I think you're honest. You didn't double spend that coin. Here's your product, feel free to leave my store with the product."
Demetri Kofinas:	32:23	That's very interesting. You're saying you can silo the network and you can branch it off and ... in this scenario you're describing with the firewall?
Leemon Baird:	32:29	That's right. They would call it partitioning the network and it allows you then to double spend. But even in normal operation, it's not technically byzantine because you never know for sure, and then this shows you why that matters, because when you can break it off like that, you have a problem.
Demetri Kofinas:	32:44	All right. So ... 'Cause in the interest of time, let's get into hashgraph because-
Leemon Baird:	32:46	Yeah.
Demetri Kofinas:	32:46	This kind of touches on the gossip aspect of this-
Leemon Baird:	32:50	Yeah.

Demetri Kofinas:	32:50	Which is that the network is communicating with everyone sort of simultaneously and explain to me this a little bit, because I tried to follow it. I read through your white paper. Again, I'm not an engineer. It was challenging. Explain for us a little bit how hashgraph works.
Leemon Baird:	33:05	Yeah. So first of all, if we're going to agree on the transactions, we at least have to know the transactions. Right? Every computer's gonna have to know the transactions and you had mentioned sharding, we can talk about that in a minute, but really it doesn't change what we're saying here. We're just talking about a single shard right now. So we're saying every computer needs every transaction, clearly, we have to get every transaction to every computer somehow. You could say, "Well what's the best way to do that?" Well we could send them all to a leader and have the leader send them to everybody, but that's slow. We have a bottleneck. So what the normal way of doing things super-fast in computer science is, you do a gossip protocol and that's the simplest thing you could imagine. If I have a transaction I've created, I just give it some random computer. I pick a computer at random and hand it to that computer and then each of us go pick another computer at random, and the two of us give it to those other computers. Now four of us know it and then each of the four of us give it to some computer at random and now eight of us know it and it just explodes exponentially fast until everybody knows it.
Leemon Baird:	34:05	This is used all over the place in computer science. It's used in bitcoin for two different things. It's the fastest way and the most resilient way we know to get a message out. So if you want all the computers to know transactions, you're going to have to get it to them somehow. This is the fastest and most resilient way you could.
Demetri Kofinas:	34:23	So as you move along this linear progression, it doesn't take much time before we reach a point in the progression of the network, where everyone knows everything is your point. Correct?
Leemon Baird:	34:33	Oh, yeah. So first, we all have to know the transactions, then the hard part is we have to come to consensus on them. And so lots of people would say, "Well, to let everybody know the transactions, just gossip them out." But then we have to do something else that is slow and has lots of communication in order to reach consensus. Instead, what I propose whenever you give somebody a message, you just attach a little tiny note to it that say, "Oh, by the way, let me tell you what the name of the last message I sent was and the last message that the last

person who talked to me was. I send you the hashes of those two messages." So let me say this again, we're constantly sending messages back and forth and when I send you a message, I attach to it the hash, just sort of the name of last message I created, and if Alice was the last person to talk to me, the last message that Alice created. That's it. I'm adding maybe one percent to the size of this message. I'm just adding the tiniest amount of information to this message.

Leemon Baird:	35:33	So we have used the world's fastest way to get the transactions out there, the most efficient use of bandwidth that you could imagine to get the transactions out there and I've added a cost of maybe one percent to the number of bytes we had to send, just to tell you the last two messages that I know about, my own analysis. At that point, we don't do need to do anything to reach consensus. Here's the magic, those two hashes ... and you can actually compress them down to just one or two bytes, for ... talk about why, they take almost no memory or bandwidth to send them. When you get a bunch of messages and you get those two hashes on each message, it forms a big graph, which is a big data structure in memory, which lets you see the complete history of how everyone has talked to everyone. You have an amazing view of how we've talked to each other, and it turns out you can run really sophisticated algorithms on that data structure in memory without talking to anyone and you know the consensus and you know that you know the consensus. And you know that everybody else is going to agree with your consensus, guaranteed mathematically, and that's where the byzantine fault tolerance, purely asynchronous all comes in. You do this with zero communication. You get it for free. That's hashgraph.
Demetri Kofinas:	36:49	And this is also massive compression, you're describing here?
Leemon Baird:	36:52	It is enormous. Well it's compression in the sense that we're sending zero bytes rather than a whole bunch of bytes. That's about as good compression as you can get.
Demetri Kofinas:	37:00	So talk to me a little about how the voting works within this. How does voting operate?
Leemon Baird:	37:05	So it turns out that it's really hard to build a byzantine algorithm. It is really, really hard and the problem was solved decades ago and there's an enormous literature ... scientific papers, academic papers on how to do byzantine fault tolerance on just a yes/no question. It's even harder if you wanna put things in order and it's even more harder if you wanna make them fair, where no one can influence that order. But we have

decades of papers on how to do this and what they say is, "Well if we have a yes/no question, we'll all send out a vote on whether we like the answer yes or the answer no better." And then you'll collect everybody else's vote and then you'll send everybody a receipt that tells everybody what everybody else voted. Now we're up to an in-cubed algorithm if you're keeping score at home. If we had a thousand of us, we're up to a billion messages going over the internet and then that just gives us one round of voting and I think about the votes and then I've changed my vote and then we do it again, and we do it over and over and over again. And then it's mathematically guaranteed that eventually with a probability of one, we all come to an agreement.

Leemon Baird: 38:05

It is mathematically beautiful. We get wonderful math proofs about it. Asynchronous byzantine fault tolerant, but it is so horrendously inefficient. To my knowledge, no one has ever deployed such a system in the real world. On the other hand, if you just do the really dumb thing that I said. You just gossip all your stuff and you add two hashes to each thing or compressed hashes to each thing you're sending, and you get this beautiful hashgraph in memory, which is the whole history of how we talk to each other, then you can run one of those algorithms that took billions of messages, and you can run it in your head with no messages at all. Zero. And so we're doing virtual voting. We're doing voting, which is decades old idea. But we're doing it without actually doing any voting. That's virtual voting. That's a new idea and the only reason you can do this, is 'cause you have this hashgraph that shows you exactly how everybody talked to each other. And you'd get that almost for free by adding just two hashes to each message, actually two really compressed hashes to each message.

Demetri Kofinas: 39:04

So the messages or the events actually function as the voting participants?

Leemon Baird: 39:09

Isn't that crazy? And the funny thing is, the messages don't even have any votes inside of them. All they have is a record of how we talked to each other. But then I can think in my head, "Well you know what? I know everyone you've talked to." And when you learn things and when those people learn things and when those people learn things, if you were to vote, I bet I could predict what your vote would be. So I'm just gonna pretend that you sent me that vote. You don't have to bother sending it to me.

Demetri Kofinas: 39:34

This is remarkable and it's also remarkable that I'm able to have this conversation on any level with you, because I must say it is

		a struggle, and I am concerned about how much our audience will follow. I had mentioned the compression before-
Leemon Baird:	39:45	Yeah.
Demetri Kofinas:	39:46	Because for me, that just seems the most remarkable, but maybe I'm missing something. But it just seems to me like there's a massive level of compression happening here in terms of the information you're able to process and you're able to actually effectively work off of. Am I missing something or is that-
Leemon Baird:	40:02	Yeah.
Demetri Kofinas:	40:02	I mean obviously compression is part ... in partial of cryptography but ... Yeah. So go ahead.
Leemon Baird:	40:07	Yeah. So it's not compression in the sense that using PKZIP-
Demetri Kofinas:	40:11	Right, right, right, right.
Leemon Baird:	40:12	Or want you to create zip files.
Demetri Kofinas:	40:12	Yes, yes, yes. Of course not.
Leemon Baird:	40:14	But use compression in the sense that what used to take billions of messages over the internet, now takes zero messages over the internet. You could call that compression and the reason we're able to do it, is 'cause we have this thing called a hashgraph that gives us the history of how we talk to each other, and that's just weird. People haven't generally built such a history and it sounds like it would take lots of memory to do it, but it turns out that you can compress that whole history down to just adding a couple of bytes per message and you're fine. So that's what's cool.
Demetri Kofinas:	40:38	All right. Let me ask you this, because again, this is obviously beyond my pay grade. I don't wanna get caught in the weeds of the mathematics and many in our audience won't be able to process it. I'm going to make available of course your website and everything else for listeners to be able to go through your white paper and see kind of how this works for anyone that wants to get deeper into it. I have some practical questions.
Leemon Baird:	40:56	Yeah.

Demetri Kofinas:	40:56	So overview, let's take your larger point at face value, which is that you're resolving certain problems that cannot remedied through blockchain as it currently has attempted to evolve. And so let's assume that this is really the preferable technology. My first question is, this is not the first time in the history of technology that a better technology has come along that has not been adopted because of various other reasons. There are stakeholders in the industry, businesses, consumers have not wanted to adopt it let's say. Are you concerned at all with that? I mean how do you see sort of the adoption for this 'cause there are a great number of people that would potentially be losers in this scenario, of you creating a non-token based system, after so much money and time has been invested in things like bitcoin and Ethereum?
Leemon Baird:	41:44	So first of all, I don't think that's going to happen. I don't think that we're going to see that people don't adopt it. The entire credit union industry of 6,000 credit unions has an organization called CU Ledger, whose purpose is to create a ledger for the credit unions and they're using us. They're building on us.
Demetri Kofinas:	42:00	Wow, really?
Leemon Baird:	42:00	Yeah, we bid out Hyperledger and other people that we'd competed with, and we have some other big customers as well. So I don't think that it's correct to say that we are ... My prediction is that it's not going to be the case that it's not adopted. I think it will be.
Demetri Kofinas:	42:12	Okay.
Leemon Baird:	42:12	Also, we haven't talked about a public ledger with a token or cryptocurrency yet and we obviously are going to be talking about that in the near future-
Demetri Kofinas:	42:20	Well that's a great question too. I mean how would you do a cryptocurrency with hashgraph?
Leemon Baird:	42:23	All you need for a cryptocurrency is some way to put your transactions in order. That's all you need and we have that. And what you really need though ... Honestly, I think that if you wanna go to micro transactions, you want really cheap transactions, you need more than seven a second. If you wanna replace credit cards, you need 10,000 a second. If you wanna go to really cheap micro transactions, you need hundreds of thousands a second per shard, and that's what we have.

Demetri Kofinas:	42:47	How do you deal with the money supply issue of finding some balance in terms of the amount of currency that can be created?
Leemon Baird:	42:53	The program that you write just is written inherently to keep a fixed money supply. So for the moment, we haven't really talked about the algorithms that we're doing. But it's ... To people in the field, it's kind of obvious how you can create a fixed money supply on a system if you have a way of putting transactions in order and we do have that. What's interesting is that there's more to the world than cryptocurrency and in addition to that, a whole new kinds of things open up like a stock market where the ordering of the bids is fair.
Demetri Kofinas:	43:21	Of course.
Leemon Baird:	43:22	Or playing video games even, an MMO could go on this because it is so fast. It's also low latency, a fraction of a second, not 10 minutes. So when you do something, you find out that it's reached consensus in a fraction of a second, it doesn't take 10 minutes. And in some experiments, it actually took a few seconds, but certainly much less than the seven seconds that a credit card approval requires and less than the 10 minutes that bitcoin is taking right now. So what's going to be important is that we've been kind of in stealth mode. We're starting to come out of stealth mode and I appreciate you having me on this program. That's part of this.
Demetri Kofinas:	43:55	Look, I am excited to have you on.
Leemon Baird:	43:56	Yeah.
Demetri Kofinas:	43:56	I wish we could talk about this in greater detail. I mean this is the first time ... I mean I spent the last few days trying to go through this. This is the first time I heard of hashgraph and the assertions are fascinating and just the basic level of which I can understand, the mathematics is impressive.
Leemon Baird:	44:11	Thanks.
Demetri Kofinas:	44:12	How long ... You know speaking to that point, how long have you been in development? Like when did you begin this process?
Leemon Baird:	44:17	Oh, I began five years ago working on the math of it and I was ... I kept convincing myself it was impossible. You can't get the strong security guarantees without a voting system, but a voting

system's too slow and if you try a hybrid system, then you get all of the vulnerabilities again. They come because of the leader that's mixed into your system. I just kept convincing myself it was impossible but it kept gnawing at me for years. I couldn't stop. I would set it aside, proved to myself it was impossible and set it aside. Eventually I realized, oh, you just add a couple of bytes to each message and suddenly you know the entire history and then you can do virtual voting, but that didn't happen for a while.

Leemon Baird:	44:48	Two years ago, we founded a company to do this. We spent a year building it up, we came out of stealth. You can go to our website, swirlds.com and download. For the last year, you've been able to download it and what we've been doing ... So in that sense, we came out of stealth a year ago, published the paper and everything, but we've been quiet. We've been quietly going to companies to get market validation and to get feedback that it's working well. We had been getting good feedback from them and now we are starting to be much more public about it. We'll be talking soon about the public thing that has the coin, all that sort of thing. But I personally been working on it for five years.
Demetri Kofinas:	45:23	When did you start sort of bringing this out because you guys have done a hell of a job with stealth? I don't know how many people are aware of this technology.
Leemon Baird:	45:30	Oh, well probably not a lot.
Demetri Kofinas:	45:31	No.
Leemon Baird:	45:31	We just have started so TechCrunch is a pretty big conference and they have a hackathon, and we just had a hackathon. There were only eight sponsors, we were one of them and we gave a prize to the team that made the best thing, was an auction site that they built on top of us that used the fairness property, that you really wouldn't wanna do with something that wasn't fair. So we just came out there. This program that I'm doing with you is here, the IEEE just had an interview with me. There's a lot of stuff we're doing but it's really all very new, being able to be talking about it publicly in this way is just a very new thing and yeah, I'm enjoying talking with you.
Demetri Kofinas:	46:02	So how do you see this playing out? In other words, how do you plan to engage let's say the financial community or how are you engaging them, developers? Are there ways for people to imbed this protocol in the software they're already developing for blockchain? I mean how would you make this transition less

painful, let's say for a lot of companies that have invested a lot of money in this?

Leemon Baird:	46:25	Oh, yeah. You asked about the financial community, that's where all of our traction has come, and we are starting to talk to other people in other industries. But it turns out, financial is leading the way in this, and their interest in ledgers in general and that's where we've been finding our early traction as well. Like the credit unions that I told you about, that's in the financial industry. Then we have other types in the financial industry we're talking to as well. So we do intend to continue with that and we'll talk about branching out to other things as well and we are ... We have this thing on our website that you can download, you can play with it, you can build apps on top of it. There's another thing as well and we don't have this, but we have been talking with people about doing this and there has been interest. But there are a lot of big systems that are being built, like Hyperledger that are being built modular.
Leemon Baird:	47:10	So the idea is that you have this entire stack of lots of pieces of software. The bottom one does consensus and then on top of that, you have just lots and lots of layers and all of these big systems are smart. They're making it modular so that it's very easy to take out the bottom layer and replace it with something else, and so we are also working with a company right now to be doing plugins for the systems so that you can plug it in as an alternative consensus algorithm. And you get the speed and you get the scalability and you get the security and you get the fairness.
Demetri Kofinas:	47:44	So how someone get an STK from you guys and like what is the licensing structure for this? How does this work?
Leemon Baird:	47:49	Yeah. So if you go to swirlds.com, you download the STK. It has a platform that's a program that's running and then it has six example apps and they run on top it. So that ... The platform does the consensus, it's like an operating system and then the app is like a program running on that operating system. The six example apps do all sorts of things, including cryptocurrency in the stock market and in a game and all sorts of things. You have the full source code for the six, you have some documentation files that explain how to write on it. What we have found ... The credit union industry actually didn't even talk to us when they were implementing their app on top of our platform. They just went off and came back and said, "Yeah. It was as easy as you said. We wrote it, we're done."

Leemon Baird:	48:26	The hackathon was really encouraging. Lots of teams only had 24 hours and they found it was pretty easy to develop on. So I'd say if people are interested in that, they should download it and play with it. If they're interested in licensing, if they're interested in deploying this in a real world use, please talk to us. Send us an email. You can get to us through the website or you can send the email or send my ... our CEO Mance Harmon, email directly and talk about licensing. We'd love to talk with you. We've got a lot of people talking to us.
Demetri Kofinas:	48:52	Give that website ... Spell the website out. Spell Swirlds out so people know where to go.
Leemon Baird:	48:57	Yeah. It's S-W-I-R-L-D-S. Swirlds. So it's sort of a shortening of shared worlds. It allows you to create shared worlds on top of-
Demetri Kofinas:	49:10	No, that explains it. I was trying to figure out what that was.
Leemon Baird:	49:12	Yeah.
Demetri Kofinas:	49:13	What is hashgraph is the website, correct, that has the information for how the hashgraph algorithm operates?
Leemon Baird:	49:18	So if you go to swirlds.com, you can see all of our papers, all of our videos. We have three videos and a podcast. Hopefully, at some point, we'll have permission to even link to this podcast. We have all our papers and those things. We have the code that you can download. You can get everything that you need there and there's a contact form if you wanna talk to us about licensing or you just have questions or whatever.
Demetri Kofinas:	49:36	So Leemon, I must say, this seems to me pretty disruptive. This technology seems pretty disruptive and maybe I'm missing something here, but it seems like it would be an unwelcome competitor for many of these alt coins and bitcoin and everyone else that has been using blockchain and proof of work and have sort of spent a tremendous amount of money in this space. I mean it's great that you are providing this solution for these major problems you're describing, not just the security issue, but the consumption of energy.
Leemon Baird:	50:08	Yeah.
Demetri Kofinas:	50:08	I mean these are huge problems, but am I missing something here in terms of the level of disruption that this is going to cause if this is adopted, which is your hope obviously? I mean

		this would come as a shock to many people that have been speculating and in the alt coin market.
Leemon Baird:	50:24	I think that the most modest thing for me to do is just to say I'll let the market answer your question for you.
Demetri Kofinas:	50:29	No, I think you're answering my question with your answer and I didn't ask it to be provocative for anyone. I just said ... Like I'm saying, I wanted to do this interview with you because when I came across your distributed ledger technology, with my limited capacity to understand it, I felt that it seemed, if in fact your assertions and your, and the community -- and it sounds like you're getting a lot of positive response from everything you said. It just seems to me completely disruptive in every facet of the term.
Leemon Baird:	51:00	Well I understand exactly what you're saying.
Demetri Kofinas:	51:02	Okay.
Leemon Baird:	51:03	Of course, there's room for more than one of anything. Ethereum didn't kill bitcoin but well, it's pretty exciting right now.
Demetri Kofinas:	51:10	Well let me ask you this then before we wrap it up. I mean are there places where you see bitcoin and Ethereum, where they can remain competitive in your ecosystem, where they're still a value add for their software and where there's a place for it?
Leemon Baird:	51:26	Let's just see what the future holds.
Demetri Kofinas:	51:30	Okay.
Leemon Baird:	51:30	I don't know what's going to happen.
Demetri Kofinas:	51:30	Sure.
Leemon Baird:	51:30	A lot of things have been built on them. Those are good things and let's just see what happens with them.
Demetri Kofinas:	51:33	That's fascinating, it's fascinating. Leemon, I wish I could've been better informed for this conversation, but I think I needed to have this conversation with you, to even begin to get there and-
Leemon Baird:	51:42	Oh, you were great.

Demetri Kofinas:	51:43	Well I hope-
Leemon Baird:	51:44	You actually understood this really well. I appreciate it.
Demetri Kofinas:	51:47	I hope our audience forgives me and I hope our audience does check this out and if they have any questions, they can certainly reach out to me or reach out to you. Is there a way for anyone to contact you or is it just basically through the Swirlds website?
Leemon Baird:	51:58	Well yeah, there's a contact form there, but you can also just send me an email. Baird is B-A-I-R-D. Yeah, if you google me, Leemon Baird, L-E-E-M-O-N. That's a fairly unusual name. You'll probably get me if you google me. But you can send me something at @swirlds.com and use also a form on the site, where you type in your message and I always respond to those quickly.
Demetri Kofinas:	52:17	So Leemon, I wanna ask you to promise that you're gonna come back on the program in the near future, after I think about our conversation and I digest it and go and do some more research. Because I feel like there's so much that we could've covered that I just wasn't prepared for us to be able to discuss in this conversation. So I hope that we can talk again in the near future.
Leemon Baird:	52:37	That'd be great. I hope so too.
Demetri Kofinas:	52:39	All right, Leemon. Have a great day. Thank you so much for coming on the program.
Leemon Baird:	52:42	Thanks. You too.
Demetri Kofinas:	52:46	And that was my episode with Leemon Baird. I wanna thank Leemon for being on my program. Today's episode was produced by me and edited Stylianos Nicolaou. For more episodes, you can check out our website at hiddenforcespod.com. Join the conversation through Facebook, Twitter and Instagram at hiddenforcespod or send me an email. Thanks for listening. We'll see you next week.