(cleaned) Participant 5 and TE Study

Nathalia Scherer 0:01
So to start, can you tell us a little bit about reporting in progress, your personal journey and how you got involved with your field of work?

participant 5 1:08
Sure. So I was raised on a farm in Canada. And on the side though, I had a computer since a very early age. So I actually I was programming since very early age. And on the farm, I was always building things to you know, fixing tractors, building green bins, all that sort of thing. So I was always building things. And by the age of 10, I knew I'd be an engineer. And, you know, I was writing computer games for fun and all that. So, you know, writing software, video games and more from an early age. And I just loved the idea of creativity and invention and all that stuff. I studied undergraduate in engineering in computer science, and then straight after that, I started an AI company this was 1999. Actually, even during during undergrad my final two summer jobs was working for national defense Canada, where I was doing AI for time series classification problems. And you know, back then this is the 90s I couldn't believe that someone would pay me to do AI and this was a very rare thing. So you know, I've been playing with AI on my own and hacking on it, but no one you know, is unheard of to get a job. So I thought that was very fortunate. So out of that I got really good at AI. I was hacking on my own but then I you know, because someone was paying me for it that I just was working 16 hour days just to learn and do good. So, from that I, you know, wrapped up my undergraduate studies with the two degrees and started my first company, you know, spent in my final year of undergrad doing the prototypes for a company that basically was AI technology that did come out of Stanford and we took something that the Stanford work and made in 2,000x cheaper to run so we did a starter for that this was actually CAD software for designing computer chips, specifically on like circuits. And I was actually doing creative AI. So but instead of needing you know, 1000 machines running for a month, we could do it in 10 minutes on five computers. This was in 1999. So we're pretty happy with that. And you know now the mainstream has got discovered all the sorts of things that computers can be creative and stuff, but you know, I was doing this in the 90s. So did the startup raise money, it was you know, height of the.com the first.com Boom. So, we you know, raised over $10 million and built the company and then sold it in 2004 to you know, one of the biggest players in the industry. From that I started two things almost simultaneously a second company also in the same space AI for chip design. In this case, solving a different problem is less of a creative AI and more about large scale verification of chips. And then at the same time I embarked upon a PhD in [...] in in Belgium, doubling down on the AI for creative design side, really asking the question, how do I leverage existing bodies of knowledge? In my case, analog electrical engineering and you know, having where the computers can leverage that while they're doing their own creative work. And in both cases, you know, the outcome was really positive for the company. It got acquired. Well, so basically, I finished the PhD in 2008. It did well, when a bunch of awards I had a bunch of publications, all this and this was publications in both the CAD field, you know, electrical engineering CAD field, as well as in the AI field, specifically, evolutionary competition and a lot of genetic programming. I always really like that

field because it's the most general possible search algorithm evolution with the most general possible search space computer programs because if you have a computer program, then you know, they can generate anything else. So I really loved that space. I still do. And that's where I spent a lot of time in the world of AI. On the heels of that then I focused 100% on the company rather than, you know, 100% company 100%, PhD, and this company was, you know, helping people with things like if you're Nvidia, you might have a team of say 10 or 20 engineers that had just designed the heart of the latest GPU processor. How do you know that it's going to yield well, right and this was really a problem, right, starting in the mid 2000s. You know, Sony famously shipped their ps3, and it was about one year late to mass production. This was because they couldn't wrap yield instead of getting 95% yield. They're getting 5% yield. So they were actually I think our third customer it was Qualcomm than Nvidia than $name recall. So anyway, you know, we built tools for them to be able to estimate yield very quickly, and then also to rapidly iterate and improve the design against the yield problems. And this is all very statistical problem, right? Like probabilities of this happening probabilities are that with anywhere from 10 variables to 1000 variables or more, anywhere from 100 samples to 10 million samples or more. So that was basically my second company and we grew it to profitability by 2010. And then, by that time, I kind of built everything I started to build. So I was starting to think about what I wanted to do next. And being a big nerd I follow all these things like augmented reality and VR and and all this and Bitcoin came on the scene. I became aware of it in 2010 2011. I bought a few Bitcoin, not enough but a few. And 2013 was when I really first discovered blockchain and really got it really grocked it.

So until then, and that's when I started hacking on what's now called NFTs right? We were building Bitcoin NFTs on Bitcoin back in 2013 and ship that in beta in 2014. And then [...] right after working with artists, etc. So, so I pulled up from my startup around that time, and, and then that startup actually, you know, I built everything I started to build, like I said, it was profitable, and in good shape. So it was just in the one day envy scaling and it was also acquired by a big company in 2017. So myself, you know, I started hacking on Bitcoin and stuff, but that led to a startup called describe, which was basically NFTs on Bitcoin. And then we were to really to early though, I mean, you know, NF T's finally took off in 2021. We were doing eight years earlier. So we pivoted to something called big chain dB, which is a decentralized database, basically. $name wrapped with with [...] BFT but that was targeted too much to enterprise and they didn't really take off for enterprise enterprise were very reluctant to adopt, they still are, because liability is the summary. So but then, you know, from my AI background, I was seeing that people were having a big challenge with data. So, you know, as an AI researcher, how do you get your hands on data? As a large company or otherwise, you have all this data? How do you monetize it? As individuals? How do you keep your data private, all these questions and so, and we saw that a good answer to this, if we really want to have there was a data economy. It was just a shattered economy, you know, opaque all that and just like, you know, the banking economy, it's been there for, you know, hundreds of years, but it's been very opaque and shadowy, but you know, Bitcoin and then Defi brought it into the open to make it transparent. So the aim was to do the same thing for the shadow data economy by making open transparent in terms of provenance and all that but still preserving privacy. And that was what became $name. So I embarked upon that in earnest in 2017. And I've been at it ever since. You know, we built what we set out to build, but now it's really doubling down on traction. And along the way, in actually you know, even in 2017 We are trying to design the token model for $name and I found that I was flailing so and I realized that I said okay, I'm gonna sit down and you know, apply, you know, my own engineering methodologies and heuristics things that I've learned from the world that circuit CAD. And from that, I, you know, basically wrote down, you know, the objectives and constraints that I wanted, and then I treated it like an optimization problem. And then I started applying, you know, solvers, which was, in this case, applying different building blocks, different token designs that existed and so on, trying different things, some work, some didn't, I would analyze against my criteria. And then from that it was almost there with just you know, one building block and with to a combination of a couple building blocks was really close. And then with tuning on that I hit it, so I was able to basically go from a state of flailing to a state of actually, you know, coming up with a design that we went with for $name much more quickly. So three months of flailing, and then realizing this sort of methodology, and then one month and done in a way that we're really proud of so I wrote about that I gave talks about it, and that, you know, helped to kick off token engineering, right? People were doing token design before, but I was one of the first people to start to put it in writing and calling it that and I called it engineering very specifically because I'd seen software engineering itself take off in the late 90s. And there's a lot of debate shouldn't be engineering should be something else. And before that, they're only computer scientists officially right. But you know, the computer scientists are competing concerned with the science side where the KPI is knowledge, not building artifacts that people can use. But people were trying to construct software that would be useful, etc. And you know, the key that's the KPI of engineering is building useful artifacts that people want to use and maintaining them and having a sense of responsibility and ethics and all that. So I realized this is exactly what we needed for for token design, not just the construction side, but also the maintenance, the sense of responsibility, all that right. So and that brings us to Yeah, so I mean, you'll probably get to it but for definition of token engineering, even in my first article that I wrote, we're actually use the term in 2017. I called it token engineering is the theory, practice and tools to analyze design and verify tokenized ecosystems. And I've actually later in the article I also talk about how that the fourth pillar of the ethics is critical as well. Anyway, so that sort of rough history and I'm still at $name, you know, working hard at it driving traction. You know, it's all about decentralized data and AI. So yeah, that's, that's what my my background is and a bit of what I you know, token engineering too.

Nathalia Scherer 11:23
Yes, thank you. Thank you so much. And I think you've already tapped into a few of our following questions. And maybe to get a bit more specific into what differentiates token engineering from other disciplines. So, can you tell a little more about what would you say that token engineering is doing and solving that other fields or not? Yeah,

participant 5 11:55
so it's useful you know, I'm an engineer, right, PhD engineer. And I've also read a lot of like thought a lot of what the philosophy of what it means to be an engineer, what it means to versus what it means to be a scientist, right. And as a PhD, you actually are trying to contribute to science as well. So if you think about, you know, if you you know, for example, what are the

different engineering disciplines out there, you know, you go to university, after our first year engineering, you you choose which sub discipline of engineering you want, right and university I was at, I could choose between electrical engineering, mechanical engineering, chemical engineering, geological engineering and a couple more. So, there was these options. And in my case, I chose electrical engineering. These days, it's typically merged with a computer, so electrical and computer engineering. So, you know, that's from the perspective of engineering. And there from the perspective of engineering and engineering is really this field about, you know, constructing these artifacts, making them etcetera. But there was a wide gap in anything related to designing tokenized ecosystems, whether they were at the level of a chain, you know, an L1 or L2, or at the level of smart contracts on top running on those chains around the incentives and also I realized that the problem you know, there was digital and analog aspects, just like there are in electrical engineering, you know, the digital side is, you know, making sure that the logic is working in the, the analog side is making sure that you know, things that are more continuous valued and value or time are working to and those go hand in hand. So, and the incentives tended to be more, you know, the analog side, the continuous valued side, I also saw, you know, there was basically by thinking about it as an engineering field. It became quite clear what the gaps were, in order to improve the practice, and people had been talking about the problems here and there and here and there, but it was scattered and there wasn't really a framework in which to hang the problems the challenges on right like what what verification which should you do, but you know, within engineering and verification is just one of these, you know, small set of checks that you do, right, like, as part of whatever design process you follow, you're going to have a verification step, right. And ongoing maintenance. And, you know, my final year of undergraduate, we took an engineering ethics class and that was a you know, a necessary class to take and the engineering that I cited, I know this is very common throughout engineering disciplines. And related to that to actually the ethics thing is super important, at least to Canadian engineering. So when I studied in Canada, and when you graduate, you get an iron ring. And the myth there is that with this iron ring, you it was the myth is that the the Iran is from a bridge that collapsed. I think it was the Tacoma Narrows Bridge or maybe another one in Canada, I forget. And you know, you get the iron from that. Now, that's not actually the case. If it was iron in the fall apart, you quickly it's actually made of, I think, stainless steel, but it's a nice myth. And so as an engineer, you're you're always dreaming of you know, as you're working hard all those years. You know, it really, really intense hours, you're dreaming of getting that iron ring, and there's a special ceremony where no one's allowed to talk about what actually happens in the ceremony. Well, then nothing weird or bad happens. It's just like this myth as part of it right? And it's very solemn ceremony. So um, and that basically underscores the importance of the ethics. So towards your question of the gaps from the perspective of engineering disciplines. There was no specific set of artifacts that were related to, you know, designing, constructing, deploying maintaining tokenized ecosystems. And then from the perspective of tokenized ecosystems, there is a whole bunch of gaps where, you know, related to things like verification or ethics or you know, design methodologies or tools, you know, CAD tool was all this just a lot of gaps or maybe there was some very nascent stuff, but there was no framework for it, but by having this very simple label of token engineering, you know, both the token side, the engineering side, then it gave a unifying framework for all this. So, and in terms of, you know, other fields that are similar that aren't engineers don't have the word engineering in them. There is this traditional field of economics in economics called mechanism design, mechanism design, however, so economics itself is a field of science, right? And so anything that is within that some fields of economics, like mechanism design is science. Unfortunately, the thing is with science, the KPI is knowledge. The KPI is not to design and build artifacts and maintain them, etc. So if you're training an economist trying to design stuff, it's sort of like you're, you know, trying to build, do your job, but with only with a toolbox or quarter full, right? So whereas if you have the label of engineering, then you actually have a full toolbox and there's training around that, etc. So that's why the label of engineering is super important. And if you look at the work and mechanism design, sure, like lots of the science, ideas for mechanism design can contribute to token engineering. That's wonderful. But it's a mistake to ever mistake science for engineering and vice versa. Right. So and, you know, all the other engineering disciplines of course are informed by science to write electrical engineering, but especially by physics, and then some fields of physics, right. Like quantum mechanics. has a big impact on designing semiconductors and electromagnetism, electromagnetic theory, Maxwell's equations, all that all of these things really play in right optics, theory all this, so and the other engineering disciplines too. So of course, the engineering disciplines are informed by by science as well as other learnings right management etc. So that would be my summary is that from an engineering perspective, the gaps were that there was nothing around this particular set of artifacts and from non engineering perspective there was, you know, sort of stuff being done a little bit here and there and here and there and tokenized ecosystems, but it was you know, with a toolbox one quarter full and no common framework to unify it.

Nathalia Scherer 18:12
Great, great. It's super clear. Thank you. And now, let's dive a little more into the specifics of the day to day work of someone working with the token engineering field. Can you share maybe some examples of typical tasks? processes that you need to handle routinely? And also specific tools that both you use it and that you see token engineers using?

participant 5 18:46
Yeah, so overall, I think, you know, Wikipedia has a has a really great article on engineering. And it's actually probably the best place to start because the just the very first paragraph and I'm gonna copy and paste it into the chat here if I can, but then I'm gonna just read it out loud here too. So from Wikipedia engineering, engineering is the use of scientific principles to design and build machines, structures and other items, including bridges, tunnels, roads, vehicles and buildings. The discipline of engineering encompasses a broad range of more specialized fields of engineering, each with a more specific emphasis on particular areas. of Applied Math, Applied Science and types of application. But let's see here, I don't so I'm looking for the design maintenance stuff because that I had to make and that's okay. The definition changed compared to what I read a few years ago, but I see it overall. And other definitions that I've given elsewhere is it's not only the in terms of the ability to practice, etc, right? There's the design of the artifact that you're looking to deploy. And within that, so I'll just give the top level things so there's the design of the artifact, there is a verification of the artifact. There is the deployment of the artifact, there is the maintenance of the artifact. And maybe that's a good summary of of the as a first cut, but then give because these artifacts, some of them could be where you want to

deploy them and they're fully immutable, right? Then they're very, very much like chips, and that's actually a good in general heuristic for token designs because if they are immutable via governance or otherwise, then you broaden the design space and governance itself exposes the contract itself to attacks by especially humans, whether intentional or not, over time. You know, the biggest challenge typically, a protocol is designed where there's you can think of it as an organization that's, you know, coordinating 100 or 1000, or a million or 10 million humans and organizations traditionally their greatest challenge not usually not usually framed this way, but the greatest challenge of organizations is capture is being caught by interests that are outside of the mission of that organization. And, you know, a good example is W3C, where they were captured by the browser vendors, and the media companies. So Microsoft and Google and Netflix etc. Basically, were able to capture it well enough such that by 2017 or so 2016. There was blackbox theorem in the browser, and you could not see what code it was. So you know, anything that's crypto with security, the fact that the code is blackbox is a security nightmare, right? And yet, somehow they managed to get it pushed through. And which is kind of crazy because you know, the way that crypto works, of course, you can have the code open source, yet still be secure. So you shouldn't have security by by obscurity by obfuscation. That's a terrible practice approach to security. It's a recipe to get insecure, going backwards to flip stack. So in this case, it's W3C is an example of an organization that was captured because and captured in this case where it's incentives became misaligned with the founding intent and in general, the mission of that organization, it's going back at a higher level right? Token engineers will be designing, verifying, deploying and maintaining these tokenized ecosystems where this tokenized ecosystem like I mentioned before, it could be an L1 chain and L2 chain. It could be smart contracts. It could be some other thing that's a combination, maybe it's a bridge that combines various pieces. And then in terms of the design, right, a lot of these practices drill down into further sub steps, right things like requirements gathering, prototyping and then, you know, there's lots of different approaches to you know, actually and then, you know, design deploy, design, build, deploy, has all sorts of ways you can do it very fast, you know, ship, a tracer bullet style design or an MVP very quickly and then iterate, iterate, iterate, or you can do it in waterfall method or otherwise and it kind of depends on you know, how immutable you want your contracts to be. You can also make it where they are immutable for, you know, the first half a year or a year and then you just throw away the key, right? There's things there's ideas like that too progressive decentralization, all these sorts of things. Lots of heuristics. So um, so in the design, it's got these sub pieces, and that's construction to verification is something that is partly still overlooked in the world of token engineering, where people are verifying from a logic perspective, via auditors partly and via tools. Like slither on that, you know, that's basically digital verification, with formal verification via formal verification and formal methods, you know, this. So this is all digital and formal methods and these other approaches have long been used in the circuit industry since at least the mid 1980s. Analogue verification. This is actually my specialty from circuit CAD. isn't as common of a thing anymore. Sorry, isn't a common of a thing yet in token engineering. And that's actually where a lot of the problems still exist, right. Many hacks and that we've seen in the last year, year and a half or DeFi type hacks, where it's about exploiting incentives, where it's, you know, do this, then this then this and this, and it's basically finding it's sort of this continued valued space that have it possible attack vectors, that someone you know, finds a way to go to. Just some very specific piece of the space, maybe a corner maybe somewhere else or, and then just goes in and wins it right? And it might be a combination of steps, right? Quite often, the first step is doing a flashing on, but then it's just the right amount of you know, flashing followed by just the right amount of putting tokens here and putting tokens there and so on. So this is a problem of analog verification, you know, continuous value in time or in value. And for that, you know, that's something that needs to be done, but it's still quite unripe as a field and token engineering. Then, and then deployment, I mean, deployment has gotten a lot better this there are things like brownie these days for Python and many tools on the JavaScript side to Hardhat etc. And then full on development suites to like alchemy and so on. And maintenance. This is something that actually also is partly overlooked. In terms of like sort of professionalism here. Where it's, it's, well basically the maintenance is sort of like a lot of it is there. There's two kinds of approaches are a few. One is the DAO where people make proposals to the to the DAO. And people the DAO people vote with their tokens or otherwise and decide for something to change or there's, you know, what's there, if there are bugs, then those get fixed. And this can be via bug reports. And that's been part of the verification thing too, right? So you know, there's good practices here and things like in $name for example, we have an ongoing bug bounty pm unify, and most of the really professionalized protocols have this right? Where, you know, the most severe bugs, it's 100k payouts right? We give pay out like that in the last couple of weeks right. So this happens. So overall, those are the four steps and myself then in terms of you know, when I'm contributing to the field, I could sort of write a lot more about it, but I realized, for me, the thing that I can do is I

my approach is simply to keep focusing on $name and then now and then if I see something that I've done that can be super useful to the token engineering community I will write about it or give talks or otherwise, right. So along the way, you know, when I was trying to verify the $name v4 contracts, I'll actually when I was trying to design the $name system of a model for sustainability, that I decided over the course of a year with feedback from a lot of people, but then I needed to get the parameters right and from that, I built my own agent based simulator called token spice. And then at first it was just pure Python, but then we wanted to verify $name v4 smart contract stuff, including how agents were behaving with with pools and all this so the analog stuff, so we I took token spice and I put it EVM in a loop and I saw that this was like a very obvious thing from my perspective. I knew this for years before because this is exactly what we did in in circuit land. And in circuit you know, for analog verification. Spice computers finally got fast enough to put spice and loop spice is a circuit simulator. Computers got fast enough to put spice circuit simulator in the loop by the late 90s. So both my companies for analogue verification, one of the recipe one of the pieces was using spice in the loop. So there needed to be a good spice for for token and it didn't exist yet in an open source way at least. So I built it and it was easy. You know, it wasn't a huge amount of time for me, and I put it out there open source and I think there's been dozens of forks now which makes me very happy, right. And it's also I wrote a book to inspire others. It's not my focus. That's why I haven't gotten started a company from it. I could have there. There should be lots of companies from it. I hear there are some companies getting started on that. And I guess gauntlet does this too, which is a closed source version like this. But to me this is great. Right? This is happening. That's one example. Another example is just writing more articles, right? I wrote an article about this top level system design called the web three sustainability loop. And that's, you know, part of the token engineering curriculum. And I think that's great, right? It's really nice to see because it's a very good general recipe for people to help make sure they design a sustainable token. Sustainable as in, you know, you can feed you know, feed your dev team when they're building but also ensure that people building on your protocol etc, can, you know, have a chance at making a living and so on to and then another aspect on the verification side to just helping people understand the higher level and you know, the lower levels, token spice and other simulators like CAD CAD, which is I view as a behavioral simulator, and that's super useful to that I've set set as much in several places. Another example is just a high level article on verification of tokenized ecosystems, right? So I wrote an article about that too. And you know, these to me are sort of building blocks. And then often I dive into very specific building blocks, too, right? In the early days, I talked, I had things like, there's a lot of interest in TCRs. But there were some flaws. So I wrote an article about that. We never ever built that into $name but people found that quite useful. And, you know, in 2021 NFTs took off, but there was tons of confusion about how relate NFTs with IP. So I wrote a series of articles on that, right? Anyway, where it's, you know, very useful to token engineers as well as NFT building teams, etc. So I tried to basically you know, whenever I can draw on my knowledge, that stuff that's obvious to me, because of my own experience, where I can write about it assertively definitively, but also write about it in a way where it can last the ages where it's, you know, clearly useful to others over time. So overall, you know, hopefully you find that useful. It's probably going to be on your question, but that's kind of how I am my approach to help him contribute to the field of token engineering is, is with knowledge based on my own practice over time.

Nathalia Scherer 30:25
Yeah, that's great. Thank you. And I remember token spice when you when you shared that and group of people within the $name. We're very excited about it. So. Yeah. And would you would you like to add anything about tools? Like specific tools, name of tools that are used in a daily basis?

participant 5 30:53
Yeah, I can. I can. talk about tools here. And so one is tools that I use, but also from a very, from a design perspective. There's a there's a big gap for the tools that I would like to see. And I actually wrote about it in detail. I'll share the link. It's within the tokens spice repo So as a quick summary, there's we need tools for design entry for verification, for design space, exploration, those three things. So for design entry, we need a really good schematic editor where you can input netlist by editing a schematic like a visual editor, where it connecting together the parts. We have examples for this from circuits from video games from system dynamics and from data pipelines. But we do not have actually, well I don't know the status of it. Now, two years ago, it wasn't there, but maybe we can have some nice things like that. No. That's one thing and another thing is on a schematic import, you know, this is another design entry tool. So it takes in a netlist. And that list is, by the way, a text file that specifies how the parts of your design are connected, and you can use it LSU and put that directly into a simulator and it spits out waveforms etc. Right. So you know, automatically importing that into a schematic. So a schematic editor goes from you know, visual to text text file. You can go the opposite way with an automatic schematic import taking a netlist and automatically generate a schematic visually

the optimization finally a waveform viewer right so um, you know, being able to see what's going on at different nodes in your schematic for from a token perspective, this is a flow of values as well as potential values right. So flow value is you know, like current flowing through a circuit, or in the case of tokens, for example, remember, I'm going to be going from eath main net to Polygon sec, right. And then there's also potential values, which is things like from a token perspective, the the amount of liquidity sitting in a particular uniswap pool or all the uniswap pools, so different potentials, their verification tools. There's some basic tools things like so corner simulation, worst case verification with three sigma verification, and I give definitions for each of those, but then there's more advanced ones too high sigma. verification. So three sigma is you know, where your yield is on the order of 99.7%. And that's, you know, most designs you're finding for three sigma, but high sigma is for a much lower failure rate. So instead of one in 1000, that's one in a billion, or one and even 10 billion and you need this in a lot of cases, right? For example, in blockchains, you need to, if you're designing a new L one, or an L zero, where you have consensus algorithm built in, you need to make sure that the probability of your consensus algorithm failing in the span of a year is one in a billion say, right like you want it to basically essentially be near zero, right? So you need to have a very rare event failure on that. And if you look at the Bitcoin white paper, the last page or so that Satoshi actually does a very simple analysis of showing what the probability of his rule is once you've done six confirmations, you know how the longest chain rule works, and that sort of Bitcoin very well in practice. So there's high seven verification then also system identification, behavioral modeling and mixed signal verification. mixed signal verification is interesting because you need to have verify both analog and digital at once. This is the case with a lot of smart contracts. And the way that you do that is typically digital simulators around a 10x or 100x Faster than analog. So the way that you can do it is you have digital verification running in the loop with an unknown like simulator all the time. So for example, token spice right now has a national loop. But you could replace that with for example, H EVM, which does fuzzing on solidity as it's going along, right? So there's things you could do there. And then design space exploration. I'll just be very quick your sensitivity analysis sweep analysis, fast sweep analysis. Local optimization, global optimization, synthesis, which is like a, like a search but across a non Euclidean space. And then you've combined it all together variation where synthesis so you're basically combining everything at once right, searching across the most general possible space and accounting for all the crazy variation at once. And this is possible, it was sort of the culmination of my PhD and I've got open source code on that too, on my personal website, the code is called Mojito. So I wrote a book about that too, back in the day. So and then there's other things too, but they can have a feel of the tools that we need, right? So we need does that way better designers or tools way better verification tools, way better design, exploration, design space exploration tools? Where are we at right now, we don't even have like a basic version of most of that stuff I described, right? It's a huge opportunity for the token engineering space. And we don't even have a basic version, but the prerequisite for most of it was an analog simulator that is accurate to the level of EVM. And that's why I built token spice so that it's much more low hanging fruit for people to build stuff on top. So token, spice unlocks the rest of this. So that's for tools for the future, for the tools of the now and token. Like I do some token engineering, but it's overall. Well, yeah, I do token engineering and I use code. I typically use Python, and EVM, and token spice or sometimes sometimes just simple unit tests. that run on a loop if I don't need full on tokens spice. So, you know, I, I know probably a dozen programming languages, but Python has been my go to language for 20 years now. And it keeps getting better and better. Also, from the AI side of what I do, it's always it's obviously the go to tool these days. So the main tools like I'm to be honest, I'm still a pure emacs guy. So it's, it's Emacs and Python, and ganache and solidity. And then you know using also the chains out there typically EVM. And then, you know, there's, you know, building blocks on top of that, right, like, whenever you're doing engineering, there's always a for any engineering field. There's a corpus of designs out there a set of building blocks that you can draw upon right. So and electrical engineering on the digital digital side, you have you know, at the lowest level things like AND gates and OR gates and then to higher and higher levels like Muxes and then floating point units all the way up to levels of CPUs and more. Same thing on the analog side at the lowest level, you have things like the transistors and then current mirrors, and then operational amplifiers. And then once you start to get larger than it tends to combine analog and digital together, things like eight analog to digital converters, digital to analog converters, radios, like RF transceivers, all this sort of thing. So um, anyway, I'm mentioning that because it's the same thing in token land. Right. So if you're dry designing a tokenized ecosystem. If it's, you know, if it's an L1 you have a choice of different consensus algorithms and within a consensus algorithm, there's many sub blocks there, you know, do you want to go BFT with you know, deterministic finality or do you want to have probabilistic finality? And then if you choose, say, probabilistic finality, are you going to use Satoshis longest chain rule? Are you going to use you know, the early Ethereum ghost rules? Or are you going to use you know, variants on that or within the deterministic finality? Are you going to use you know, BF T are you going to use something else? And you can drill and drill and drill in and same thing for a smart contract, you know, are you are you going to have an AMM building block, or are you going to have or maybe a exchangeability block but you want that to be centralized or decentralized? Or some mix in between then if it's decentralized, do you want it to be an amm or do you want it to be some other thing? Etc, etc. And there's all these sub choices to have, right? Do you want it to have you know, weights on it the way that balancer does or do you want to do you want to have it where you can set a range of like liquidity providers can specify a range that they provide liquidity for like uniswap. So there's all these different building blocks and you know, a quick summary is even just with DeFi, you've got stable coins, loans, insurance Dexs, is these days options like derivatives and options. And if I have one more, right, and then beyond that, you've got other building blocks too. You've got TRRs you've got Bonding curves with those are really good for curation and more, and many more things. Right. And in fact, in my original token engineering article, I laid out the building blocks at the time. So that's a really a key piece. And so then to your to your question of what tools do I use day to day? Well, I have these in the back of my head, right? I'm you know, I use Twitter. Twitter's a key tool to learn about the new tools to learn about developments. So for example, in the last couple of years, privacy tools have become way better, right? You know, zero knowledge stuff. You know, we now have zero knowledge tools for L2s although most of them are you just using the zero knowledge aspect just for scaling but not for privacy. But there are some you know, zero knowledge chains [] coming along and there's more obviously, Z cash back in the day or Zed cash, as we call it in Canada. And then, but besides that, you know, in privacy side, there's building blocks, like differential privacy and homomorphic encryption. $name has something called computer data, which is bringing the computer data and, and more right so with those building blocks, you can use them in smart ways with smart contracts, right? So for example, zamanda AI, has been doing some really awesome releases around encryption in the last half year, a big release just three days ago, and that plays super well with smart contracts. So these are all building blocks in the design and you know, we are using these as tools to so you know, my my tools are the tools from a software developer perspective, from a blockchain development perspective, from you know, drawing on the building blocks perspective, and more, but I tend to operate at the lower level, you know, I rely on others for the front ends and stuff. I work with them. On the prototyping. But there are tools like overall when you're trying to deploy, you know, a tokenized ecosystem or even a very specific product within that. There is, you know, the front end, which typically is a web app or a mobile app. There's the back end, which has several components, especially the smart contract on a chain typically, but there might be other back end components too. So anyway, those are some like tools. Nothing super fancy. But that said, there's so many things I use implicitly right, um, you know, like I use coin Gecko to identify addresses in a secure way. I use ether scan all the time. I use Metamask. I use Tresor, I use a bunch of other wallets, too. So there's so many things that I just do without thinking about it, that it's just there, right?

Nathalia Scherer 42:09
Yes. Alright, so we covered a lot of a lot of ground with the pools. Thank you. Thank you for that. I think that's, that's super helpful. And we have about 15 minutes left. And I think I want to be zoom out a little bit. And you worked with AI and we see that AI continues to advance and of course, there's potential for for to significantly impact the development and implementation of TE. So yeah, how do you see AI affecting token engineering and your role in the landscape and also keeping in mind that I still want to ask you a couple other questions after that.

participant 5 42:55
Sure. Yeah, so. So first of all, I do not envision it to improve governance. So maybe I will just want to say this up front because governance itself as I've already stated, it's a security you know, humans making decisions on smart contracts, increases the surface attack area. And so you know, my ideal governance is no governance, you know, Rai is a really great example there, you know, ride or die and basically remove the governance and replaced it with a PID controller. And that was, you know, invented by a chemical engineer on surprisingly right. I mean, so many, it's a chemical engineer. You don't normally think of him like that, but he is right. Let's see a nutrient engineer. So in general, them just as you know, governance by humans is hard governance. And because you know, humans are opaque and squishy, and can be corrupted. Well, AI is opaque and squishy, and can possibly corrupt it too, but it's even harder to understand and manage. So, you know, I think examples like that, like there's, there's definitely to me, things that are useful to go for and things are not, but I want to focus on the things that you know, where AI can be helpful. So, overall, as we go forward, you know, there's this field of science called cybernetics Right? Which was invented in the 50s by Norbert Wiener, and then he actually, his wife sort of doing some crazy stuff, and it made both him and his wife very, very unpopular among academics, so people just stopped citing him. But all that stuff basically then

got reinvented over the next 20 years, but the original source of all that stuff is Norbert Wiener, cybernetics at work of the 50s and early 60s, right? So, cybernetics is a really important word because it's this idea that combines many types of systems, right? So, and it's relevant for this question here. Because future systems are going to be combinations of straight up, stop playing, you know, traditional software technology, plus blockchain token technology plus AI technology, right? And they're going to be lots of mixes and matches and overall, you know, they will be truly cybernetic and you know, there's an IEEE journal called systems IEEE Transactions on systems and cybernetics, there's actually three parts to it ABC. And that's actually been it's a long running journal more than 20 years. And it actually is about systems that combine these right so I envision that it's going to be not just one or two narrow places on the combination of AI and token, it's going to be everywhere, right? AI is gonna be a feature in a lot of places. You know, token as a feature in a lot of places. And sometimes things are token first sometimes things are AI first sometimes things are mobile first. Sometimes things are saying mobile first and AI first but token second, all of that right so we're gonna see a lot of different combinations. And I think, you know, that's exciting. There's examples out there already, you know, like the sort of the leading protocols in blockchain land at the intersection of blockchain land and token land are instructive right? There's four projects I'd say that are sort of have been at it since 2017. That are the leaders. You know, there's $name I mentioned that, you know, it's about decentralized data exchanges unlocking data for use by AI. There's singularity net, which is at the heart of it really trying to go for AGI but along the way, has built a lot of really great agent based AI technology, etc. There's fetch AI, which is also an agent based AI technology. And there is numerai, which is less web three, it uses token second, and it's much more for financial AI, right. And maybe I'll just point drill into numerai for a second because it's, it's useful to think about so it's, you know, main customers are Wall Street investors who basically invest in numerai as a hedge fund, right, but then numarray has to generate returns. And normally a hedge fund has a team of, you know, five or 10 or 20, super smart data sciency people called quants, right? And they try to you know, predict better and trade better and all that or if they have really fast computers, they try to front run everyone, right, the original Flash boys, right? But you know, these days, you know, AI and for the one for a while, AI has been a big name of the game there. So when numero dos, it says instead of having five or 10 people internally that are doing this over to the crowd right? Make it where you incentivize 10 or 100 1000 different data scientists to submit their own predictions where the market will go, and then and then aggregate that and then use that aggregation to trade against and make money as a hedge fund and they're doing really well as a hedge fund. Their numbers are amazing. So that's an example printing, right? But then, you know, the data scientists themselves making these predictions, it's all AI but it's their only purchase AI right, you know, AI models, etc. They get to stay private, etc, etc, etc. So that's an example of something that's out there already. There's many more than like token spice itself, you know, within token spice it's agent based simulation, right? Some agents can be pure Python. Other agents are a Python wrapper of a smart contract running on EVM. And so you could have it where some of those agents as you're developing something you can have, you know, an agent that's wrapping chat, GPT. You know, I call it the chapter GPT you can have agents wrapping LLMs you could have agents that are just you know, simple neural networks or simple support vector machines or otherwise. So you can have aI under the hood there. And you can also use you know, AI to optimize things people are doing this you know, gauntlet for example. And and others are using AI to well, it's not even necessarily AI. It's more just pure optimization, to optimize, trying to minimize incremental loss by wiggling weights on balance pools, such that it's minimizing permanent loss for the LPs. So there's lots of examples and you can get crazier. Crazier yet right? To me, you know, in the medium and longer term. I'm really excited about AI DAOs. And these I wrote about I wrote about this in 2016 when DAOs hadn't acquired this more narrow definition that has now and I really prefer the broader definition. How I see a DAO is its simply a group of people that is being coordinated by blockchain technology and the blockchain technology that is coordinating it. Bitcoin is a DAO. It has been you know, Bitcoin itself. is coordinating hundreds of of serious miners, 1000s of miners, hundreds of millions or 10s of millions, if not hundreds of millions of Bitcoin holders towards its singular objective, which is to survive by convincing people to mine and then paying people and so on. Right with all this speculation around. You know, Ralph Merkle has gone even further and posted on my forum which I actually largely agree with to but it's also a DAO right? And I call that a protocol Dow it's you know, Dow that's coordinating humans via protocol. You can also have a Dunbar DAO where it's just a bunch of people where they're basically collectively managing a shared wallet, but it's really hard once you get past 150 People the Dunbar's number, so, because, you know, a lot of these DAOs have this idea that they have to be flat not hierarchical. And you know, if you have a flat organization and it's yeah, 150 people it's the maximum typically 50 people is the max actually.

Nathalia Scherer 50:40
Yes sorry to interupt. But we're coming close to the end, right. And I'm curious if we go a little over, is that okay with you or want me to

participant 5 50:50
actually I have a hard cut off, but basically the idea the idea was to wrap up here is simply imagine that the smart contract is controlled by AI. Or imagine this smart contract super dumb, but the only people engaging with it are AI bots, not people. Or imagine you just have a whole bunch of agents running around, and a bunch of people running around as sort of these independent entities. Some of these agents are true people. Some of them are AIs, and they're just on the team doing nothing. So anyway, that's the idea of AI DAOs. I also writw about that. So please, yes, we have seven minutes. So please go ahead.

Nathalia Scherer 51:24
Thank you. And very quickly, what areas of knowledge do you think are key? Central for token engineering?

participant 5 51:35
The ability to to learn more than anything, you know, I know that you know, $name and Schuurman have this butterfly and I think the butterfly is cool. At at the same time, you know, you don't have to, I think it's got what eight fields. You don't have to learn all these eight fields in order to do token engineering, just like you don't you don't have to be an expert in a field to go and be an electrical engineer, right. You learn a little bit of each as you go along. And the best way to learn actually is to a project to try to build something to try to build some artifact right.

This is you know, so to me, the ability to learn efficiently, the desire and then to actually build right so token engineering like if you're the the goal of an engineering is to design construct deploy and maintain artifacts, software artifacts, otherwise, then you've got to be good at the construction side to right you can't just throw it over the fence to a, you know, in some contractors to build you the thing. You've got to be good at it yourself. So um, and only by doing the building are you learning and truly getting token engineering, right. So the most important thing is is you know, the ability to learn the desire to learn, and the ability to build and iterate right, everything else follows from that. Right. So in terms of the fields, obviously, it's not dissimilar to existing engineering disciplines, right? You need to have you know, ideally you have a good grounding and in math and physics and you and dynamical systems in particular, you should be able to program in so you know, Python and solidity or a solidity or you know, basically some smart contract language and some wrapper language, right. So smart contract language, the main one is solidity, but it could be rust or otherwise. And in terms of a wrapper language, the, it tends to be Python and JavaScript, right? You need to be able to do that and you need to be able to work around around software systems. You need you know, if you truly want to be an engineer, you've got to have the ethical bias. So reading the books, and really caring right, having the background there. And then, because you know, these systems tend to be around coordinating humans. It does help a lot to understand aspects of psychology and social science to right. So there's aspects there. I mean, basically, you you'll see the list that you know, $name and Shermain listed, and I don't disagree with that at all. I think it actually makes sense. Just just at a macro level, it can't be sort of theory first have to become an expert. It has to be rolled with sleeves and start, you know, have a project in mind, roll up your sleeves and start building and learn as you go. Right. And it really, really, really is important to build and deploy and build and deploy and learn.

Nathalia Scherer 54:28
Absolutely. Thank you. Thank you so much. And last question. Whose work do you admire in the field? And also, who do you recommend that we talk next? Considering that we're also trying to include different, different perspectives within the study?

participant 5 54:48
Yeah, sure. So um, I think, you know, I mean, I've mentioned $$name$ and $name$ on definitely, they're on my shortlist. And I'm sure everyone should listen to them to another early token engineer who hasn't done as much in the last while is $name$. He invented the $bonding curve$. He co invented $ERC 20$ He did a bunch of other stuff. So name is you know, he's brilliant and has really great insight. You know, he was co- building. $name$, which was the first you know, Blockchain music platform back in the day. These days, he's writing sci fi novels, envisioning the future but inside them, they're filled with $mechanism design$, right? Which is kind of awesome. So I think it's a really great way to for people to help learn. You know, $name$, she's great too. And, you know, she's done a lot of Awesome contributions. And yeah, there's many more. There's a lot of people involved into engineering in the last few years that to be honest, I haven't followed us closely because I've been so heads down and in building I haven't gone through a lot of events and stuff. So I'm, I'm a bit more naive than that. But I do follow a lot of building blocks. So I think it's useful to look at you know, the leading DeFi

protocols or the other protocols that have taken off and then ask who designed this thing? Who built it and then talk to them? Right? So, you know, RAI is a great example. So $name$ You know, he's a brilliant token in there. I don't know if you've ever called himself that but definitely right. The inventors of the TCR I mean, it was on that list as well as the name is right. $name$ and I forget the other name. They went on to do $name$, which was, you know, token gated. Discord and token gated Telegram, right, has millions of users, right, and they're just totally flying under the radar. It's amazing. And then, you know, also $name that has brilliant token engineering and you know, that's a spinoff from $name and led by $name$, who's also a brilliant token engineer and his team. The $nameguys are awesome in token engineering, too. They're less famous, this guy in $name$ called $name$, and he's one of the most brilliant token engineers on the planet. He's designed most of the tokenized systems within what they have done including my favorite which is $redacted cartel$, which is a fork just look at it. It's got the $Ve33$ stuff going but it's really amazing. And then also, you know, look at you know, the people who did $name$ a $name$ who did have a I'm not sure who was doing the unit token as I'm actually $name$ was the one really pushing $AMMs$ as $Dexs$ and he got inspired by someone else. And then he was nudging $name$ and $name$ ended up nudging $name$ So, $name$ is also a brilliant engineer and he was one of the key minds behind the $name$ etcetera. Also very under the radar, right although he's doing a lot more these days to drive $name$. So you'll see all the design decisions, the technical token design, decisions of $name$ are like immaculate, it's amazing the $name$'s behind all of it, right. So

Nathalia Scherer 57:48
sorry to cut you. Time,

participant 5 57:52
I'm sure look at the next time, right. So yeah, for defi. But this is the place to look right. People have actually rolled up the sleeves, design this stuff, built it and shipped it right.

Nathalia Scherer 58:04
Wonderful. Thank you so much. This has been incredibly valuable. And our plan is to continue interviewing for maybe the next month, and we might reach out if we have any questions or early insights to share. Is that okay? With you?

participant 5 58:19
Yeah, sure.

Livia 58:22
Thank you so much, participant 5.

participant 5 58:24
Great. Yes. And happy to help. Yeah, good. luck with your work.

Nathalia Scherer 58:28
Thank you so much.