EPISODE 1283

[INTRODUCTION]

[00:00:00] JM: How does crypto affect the world of gaming? Well, there are three key innovations right now. There are tokens, there are NFT's, and there are zk-SNARKS. And I actually didn't know that zk-SNARKS were applicable to gaming. But in today's episode, I was convinced of that. Dark Forest is a game that is built around the Ethereum blockchain, and it uses zk-SNARKS to create hidden information dynamics. And the reasons zk-SNARKS are useful are because they are a way of validating that you have a certain piece of information without revealing that piece of information itself.

I'm not going to pretend that I know how zk-SNARKS actually work. I've done one or two shows on them. I was completely confused. But what I am very curious about getting familiar with is how you actually use them in consumer applications. And Dark Forest is one of the most popular consumer applications that actually uses zk-SNARKS on a regular basis.

Brian Gu is the creator of Dark Forest. And he's a profound thinker about games. I think you're really going to enjoy this episode if you like gaming, or you like crypto, or you like the intersection of the two. I certainly had a great conversation.

Our first book is coming soon. *Move Fast* is a book about how Facebook builds software. It comes out July 6, and it's something we're pretty proud of. We've spent about two and a half years on this book. And it's been a great exploration of how one of the most successful companies in the world builds software. In the process of writing *Move Fast*, I was reinforced with regard to the idea that I want to build a software company. And I have a new idea that I'm starting to build. The difference between this company and the previous software companies that I've started is I need to let go of some of the responsibilities of Software Engineering Daily. We're going to be starting to transition to having more voices on Software Engineering Daily. And in the long run, I think this will be much better for the business, because we'll have a deeper, more diverse voice about what the world of software entails.

If you are interested in becoming a host, please email me, jeff@softwareengineeringdaily.com. This is a paid opportunity. And it's also a great opportunity for learning, and access, and growing your personal brand. Speaking of personal brand, we are starting a YouTube channel as well. We'll start to air choice interviews that we've done in-person at a studio. And these are high-quality videos that we're going to be uploading to YouTube. And you can subscribe to those videos at YouTube and find the Software Daily YouTube channel.

Thank you for listening. Thank you for reading. I hope you check out Move Fast. And very soon, thanks for watching Software Daily.

[INTERVIEW]

[00:03:13] JM: Brian, welcome to the show.

[00:03:15] BG: Awesome. Thanks so much for having me here, Jeffrey.

[00:03:17] JM: You work on Dark Forest. Dark Forest is a game that is tied to the cryptocurrency ecosystem. And my first question is around crypto and gaming holistically. So, gaming is one of those really flexible technology domains where whenever something new is built, whatever new kind of technology comes out, it's going to be used somehow in gaming, because gaming is such a big domain. What are the applications of crypto technology in the gaming industry?

[00:03:54] BG: Yeah, for sure. That's a great question. I think that there's a couple of exciting things about the intersection, particularly of crypto and gaming. The first thing I think is actually a little bit more general and broadly applicable to platforms beyond just crypto. I think that gaming is a really exciting opportunity to test out new technologies in environments that are often – You might consider them lower stakes, then, for example, like there's a lot of financial applications being built on Ethereum, and those things are really mission critical. And you cannot screw up smart contracts or the infrastructure that you're building around that. But gaming allows us to be a little bit more playful and to experiment with different sorts of mechanics and different sort of affordances of the technology that might be more experimental. So that's sort of a broad answer to why I think gaming is exciting in any up and coming platform,

specifically the affordances that I think are interesting around gaming on crypto, though, are I would say that the most obvious one is that you can start building games that take advantage of native API's into money.

So, the idea that, in a game, your assets can have real value is I think something that people are starting to pick up on more and more. And there's this idea almost that you can have these games that have real world consequences. So for example, in Dark Forest, if I conquer your planet, or if I take one of your items, then that might be the equivalent to me taking some sort of economic or like financial value from your player in the game.

But the thing that I think is actually even more interesting and important than this is the idea that games built on decentralized systems are infinitely interoperable and composable in ways that games built in the traditional way or not. So, for example, one thing that you get for free whenever you build any application on blockchain is that the application is client agnostic. In other words, you're going to deploy some smart contracts up to the blockchain. And that will basically define how you can programmatically interact with the data layer of the application. But beyond that, usually developers might provide like a default client to actually make those interactions. But really anybody can spin up an alternate web client of their own. We can go even deeper into beyond just the user-facing client that people are using to interact with the game. People can start writing interoperable smart contracts on top of the game defining their own mechanics, expansions, all sorts of interesting things. And we're seeing some of the early signs of that right now with Dark Forest, which is what gets us really excited.

[00:06:20] JM: Okay, so you laid out some really cool ideas there. And when I look at the ways that these ideas are actually being captured, I don't really see many companies that are doing any of the things that you're describing. I see there are people who are doing interesting things with NFT's. There are people that are basically porting old world games to have some contrived crypto mechanics. But Dark Forest is, to my mind, the first game that really integrates crypto in a unique, like category reinventing style. Are there any other games that are really taking advantage of what you can do in the crypto world?

[00:07:08] BG: I think that in the next couple of years we're going to start seeing more and more games doing this. So one thing that I'm also really excited about right now is we're

exploring with a couple of teams, the idea of other sorts of zero knowledge crypto mechanics that can be used to make crypto-first games. So, for example, I think that zero knowledge cryptography in Dark Forest is really important, because it allows for these sorts of like – We use it for these information asymmetry fog of war sorts of mechanics. But zk has tons of other applications as well. It could be used to, for example, create like a very compelling trading card game on the blockchain. It could be used to create like open sandbox worlds assuming once performance gets a little better and things get optimized.

I think that I would broadly agree that most crypto gaming experiments that I've seen in the space so far, to me either feel like take something from the traditional gaming paradigm and put some crypto integration into it. Like now the items are NFT's or there's a crypto wallet somehow in this mobile game. Or they're things like you start with some crypto application and you gamify it. So you'll see a lot of DeFi apps that are financial applications, but they feel gamified, and players often feel like using the application is very playful. But, yeah, I think I would agree that the next step for crypto gaming is really to have games that are natively built for the blockchain.

[00:08:26] JM: Dark Forest is a kind of a complicated, intimidating game for people who are new to it. I'll say that from personal experience. Why didn't you make a game that's more accessible?

[00:08:42] BG: Yeah, that's a great question. And honestly, my answer to this is that, from the beginning, at the very least – Like at the very beginning, when we started building Dark Forest, it was more of a technology experiment, at least for me personally, than it was a game. And over time it evolved into being something that had compelling gameplay mechanics for some subset of players. But initially, when we started thinking about kind of like the underlying crypto constructions for Dark Forest in 2019, I myself was just getting into zk crypto and exploring the space and seeing what was possible. And then over time I realized like I wasn't able to get – I was just so excited about the idea of actually trying to bring it to something that people could play, that we ended up just sort of layering on more and more mechanics. And I think that the core mechanic, like this information fog of war, is something that is probably more hardcore, competitive, player friendly than casual player friendly. But I think that crypto gaming will go mainstream once those casual sorts of gameplay modes exist. And I'd love to explore that more as well.

[00:09:44] JM: Explain what your game is. Like explain what the player is actually doing.

[00:09:52] BG: Yeah, so Dark Forest is basically a massively multiplayer online strategy game. It's space theme. So it's a space conquest game. Essentially, you initialize into this massive like infinite procedurally generated universe with hundreds or thousands of other players. And you spawn on what we call your home planet. Now the universe is this giant expanding disk that contains within it millions of planets. And your goal is sort of to explore the universe and gradually conquer more planets and expand your space empire. And usually, for each round, we put in some sort of objective into the game to make it competitive, to make it so that players kind of have something to compete on. In the last round, for example, the goal of players was to find certain kinds of celestial objects, which we called asteroid fields, to mine resources on them, and to ferry those to trading posts and pull those out of the game that way. And that was their objective function, and the players would be scored based on how well they were able to do that. So you're kind of exploring the universe, growing your empire, completing some objectives, competing with other players, engaging in diplomacy with them, space conquest RTS game.

[00:11:02] JM: What does Dark Forest referred to?

[00:11:05] BG: So, Dark Forest, the game, was actually inspired by – If any listeners are familiar with Liu Cixin's Three-Body trilogy, the second novel in that trilogy is called the Dark Forest. And that itself is a reference to Dark Forest theory, which is kind of a thought experiment that's an answer to the Fermi paradox. I don't know if I want to spoil it for anyone who hasn't read the trilogy and has been meaning to. I don't know. What do you think, Jeffrey?

[00:11:30] JM: I haven't read the trilogy yet. You can spoil it if – You can say spoiler alert. And I don't personally – Spoiler alert. Now's your chance to turn off the podcast if you don't want to ruin one of the best books in history.

[00:11:44] BG: For sure. So, yeah, basically, the idea behind Dark Forest theory is it's an answer to the question of why have we not been contacted by any other alien civilizations. And Dark Forest theory posits that it's basically incredibly dangerous for you, as a civilization, as an intelligent civilization in the universe, to make information about yourself known to the rest of the

universe. And the idea here is basically that suppose that you learn of the existence of some other civilization. Now, because there is such a large distance and cultural gap between you and this other civilization, it's hard to communicate, it might be hard to come to alignment with them on their values, you might not know are they a friendly civilization? Are they a hostile civilization? And furthermore, because the distances in space are so vast, you also have a lag time of – Because of limitations on the speed of light, you might have a lag time of years, centuries, or even longer on what is the latest status of the civilization with respect to technological development.

So given the uncertainty and just like the fear that you might feel around the other civilization being hostile, and the lack of information that you have, Dark Forest theory posits that the best thing that you can do is pretty much to make a preemptive strike. So if you can imagine, there might be thousands of these different intelligence species floating around the galaxy. They're all kind of – It's as if they're these like sneaky animals kind of creeping around a Dark Forest trying to not make others aware of them. Whenever they become aware of some other species, they might try to take some hostile preemptive action in order to ensure that they themselves don't get attacked first. And the idea here is basically that like information becomes a very important resource in a universe such as this. And that's kind of what the Dark Forest game is all about as well.

[00:13:35] JM: So, Dark Forest uses zk-SNARKS. And, to me, of all the innovations that crypto brings, it is not intuitive to me that zk-SNARKS would be useful as a game mechanic. Could you briefly explain what zk-SNARKS are? I think, actually, probably a lot of people listening know kind of what they're used for. But maybe we could just rehash what they're used for real quick, and then talk about their applications in gaming.

[00:14:06] BG: For sure. So zk-SNARKS are basically a cryptographic tool that allow you to prove that you have correctly executed computation on some hidden inputs. So maybe there's some computation that I want to perform, like transferring some coins between balances, but I want to keep some particular inputs to that function that I'm executing private, for example, maybe like the sender and the receiver, or like the amount of coins that I'm transferring. So what I can do is I can present you with the output of this computation, the end result of the execution of this function, and this zk proof, a zero knowledge proof that essentially acts as like a

signature on the computation. And without knowing the inputs, or the exact computation steps that I took, you can just look at that signature of the computation and, in very short order, verify it to ensure that you can trust the results of this computation.

So I think that the first two things that a lot of people get excited about in terms of applications at zk-SNARKS are around privacy technology and scalability technology in blockchains. I think the application to privacy is fairly straightforward, because essentially what zk-SNARKS ensure is that, I like by default, everything on a blockchain is public. That's what makes Bitcoin and Etherium trustless. Anybody can basically go through the entire transaction history and verify that everything was executed correctly up to the present point.

There's a downside to this though, which means that all of your transaction data, all of the inputs and outputs of anything that you're doing with any given account on the blockchain are public. So zk-SNARKS would basically allow us to perform transactions in a more kind of private manner, where we might be able to partially say like hide or encrypt certain key parts of data that we don't want to be known to the world while still maintaining the security guarantees and those verifiability properties that are so important to blockchains.

Now, the second application that I think a lot of people are really excited about is scalability. And this comes up in new sorts of layer two blockchain constructions that use zk roll up. And the idea here is basically that using zk-SNARKS, you can roll up computations into just verification. So this is actually not using the zero knowledge or the hiding property of zk-SNARKS. But what zk-SNARKS say is that like if someone was willing to generate a proof that they performed a computation correctly, in order to verify the correctness of the outputs of that computation, rather than doing the whole thing yourself, all you have to do is verify the proof. And it turns out that proof verification with zk-SNARKS is actually quite efficient. So this speeds up a lot of operations that ordinarily would take a lot of people doing a lot of redundant computation to begin with.

Now, the last application is gaming, which I think is actually quite underlooked, which is obviously what I'm personally most excited about. And the idea here is that, within games, you can sort of break up all games into two categories, complete information games, and incomplete information games. Complete information games are games where all players know the full

state of the world. So you can imagine games like chess or checkers, where like I know where your pieces are and you know where my pieces are.

Now, incomplete information games are basically all other games. You can imagine like poker, for example, as an incomplete information game, because we don't know what cards each other has. And Starcraft is an incomplete information game, because until I have a vision or knowledge of where your base is, I don't know what units you're producing or, or what you might be strategizing. Now, it turns out that like I think pretty much all massively multiplayer online games have some sort of incomplete information component to them. And I think that's because incomplete information allows us to dig into a lot more nuanced social mechanics. So incomplete information enables things like deception, or conditional coordination, and a lot of emergent sort of social dynamics.

Unfortunately, because of the open and transparent nature of blockchains and decentralized systems, it's been really hard to build incomplete information games on these decentralized systems. So, for example, you might be able to build a game like CryptoKitties where everyone knows who owns what kitty and what all the properties of each kitty are, but you're going to have a hard time going past the complexity of something that looks like a trading card game, I think, unless you can use incomplete information in some manner. And this is basically where zk-SNARKS come in. Because zk-SNARKS allow you to perform computation on private data, but in a verifiable way, they're going to open up a lot of these kinds of incomplete information mechanics that previously weren't possible because of the transparent and open nature of data on a blockchain.

[00:18:38] JM: Now, I'd like to make a little bit clearer what you just said, the tail end of what you just said. You're talking about, if I understand it correctly, cryptographically verifiable private information, in-game private information. So maybe that you could cryptographically verify that your whole cards in poker are not being exposed. Am I understanding that correctly? Is that an application?

[00:19:07] BG: Yeah. So what you might be able to do is you might be able to draw cards privately from a deck. And essentially, ordinarily, without something like zk-SNARKS, you might have to reveal your cards out into the open to ensure that later you don't cheat and play a card

that you don't actually have. But with something like zk-SNARKS, you'd be able to draw cards into your hand. And then in a future round, when you want to play a card, you can reveal one of the cards and use a zk proof to demonstrate that all of those actions ago, when you drew cards, you did in fact draw that one card, but at the time the other players wouldn't have known.

[00:19:43] JM: Okay, this is so hard for me to grasp, because like if the code is closed source, this is impossible to do. And if the code is open source, then why would you need the zk-SNARK? You know the series of computation steps that are going to be applied, right? Like I still don't understand what the zk-SNARKS solves here.

[00:20:07] BG: Yeah. This is a great question. So maybe, as an example, I'll take kind of a toy version of the mechanic that's going on in Dark Forest. So let's imagine that we have a game where location information is private. So maybe you have some pieces on a board, on a really large board with other players, but you don't want other players to know where your pieces are. So something like Battleship or something. And imagine also that you can move your pieces around. So this might be a variant of Battleship where you've got submarines, and you can move submarines underwater. But you don't want other players to know exactly where your submarine is. What you're going to do is you're going to be storing locally this private data about the actual coordinates of, let's say, your submarine, right? And you're going to upload a commitment to this private data up to the blockchain. So you might hash these coordinates, for example, and then upload that hash to the blockchain. But you're not going to upload the coordinates themselves. That's going to be stored, i.e. like if this is browser game in like your browser's local storage.

Now, whenever you want to make a move, what we need to make sure on the smart contract side is that you're not, say, teleporting your submarine across the map in a way that's not allowed by the rules of the game. So what you'll do is, locally, you're going to update the coordinates of your submarine from, let's say, 1,1 to 3,2 or something like that. Then what you'll do is you will publish a commitment to your new private state, i.e. like the hash of the coordinates 3,2 up to the blockchain. And because hashes are hard to reverse, anybody just looking at that hash, let's say it's like a salted hash, anybody looking at that isn't going to be able to reverse engineer your private data, which is the location of your piece, the submarine. And

then, in addition to that commitment, you're also going to publish a zk proof that there was a valid state transition between your first private state and your second private state.

So for example, let's say that in this submarine game, submarines can move like knights in chess. So they can move in L shapes. So you're basically going to publish a proof to the blockchain that I moved by submarine from secret location A to secret location B. I'm not going to tell you what secret locations A and B are. But this proof does prove that the move was an L shape. I'm not going to say anything about the orientation of the L shape or anything like that. Just that it was indeed a valid L shape and a valid move.

[00:22:27] JM: Okay. But why do you need crypto for this? Why doesn't the game server – I mean, I know that this would be appealing to centralization, because there's some game server. But even if the game server is a set of eth contracts, like that does the trick, right? I don't understand why you need the proof.

[00:22:50] BG: Well, what's going to happen is like, because I'm hashing my private state, I'm uploading a commitment to my private state from which my private state can't be inferred, right? So if I'm sending some hash of those coordinates 1,1 that my submarine started on to the contract, and then, without zk proof, I decided I wanted to cheat and move my submarine all the way across the map to say the coordinates 10,10. All the contract would be seeing would be like the hash of the coordinates 1,1 and then hash of the coordinates 10,10. But it wouldn't know that like those hashes correspond to those coordinates. It just sees these hashes as essentially random strings. So the contract wouldn't be able to verify that there was a valid state transition between the first private state and the second private state.

So the contract in any outside observer, these just look like random strings of bytes. And they have no idea of checking for the validity of like, “Is this player trying to cheat me? Are they trying to do something illegal?” That stuff has traditionally either been done by either making all of your state public, or else you just wouldn't be able to verify it. But zk proofs allow us to do this in a way such that your privacy is preserved. But also the contracts can check the validity of your state transitions.

[00:24:04] JM: And just to help me refine my understanding of this, like I'm starting to get an intuition for what you're talking about here. But why couldn't I just implement the game so that the client prevents me from making an illegal move? Like shouldn't the client just be able to reject any move that does not have a valid L shape to it?

[00:24:29] BG: Yes. So this is a really great question as well. And it goes back to the property of blockchain games and I think decentralized apps broadly, that is that they are client agnostic. So I think know, with a lot of traditional games, let's say like RuneScape, or something like that. It's often the case that the client that is distributed by the game studio is the game. So, in particular, the game studio will make a contract with players essentially saying that if you decide to try to use like a hacked client or an open source client with like illegal plugins or modifications to it, then that is a bannable offense.

Now, with blockchain apps, blockchain apps are permissionless. So you can't gate keep the apps by specifying which clients the users have to use to connect to the application. Anybody can make any transaction on the blockchain without having to go through, for example, our web servers that are distributing our like sort of vanilla client. So that means that anybody can, for example, use automations as they please. They can fudge transaction inputs to be whatever they like. And it's on us on the smart contract side to ensure the validity of any transaction or move that players are trying to make in the game.

[00:25:42] JM: That idea of the client agnostic architecture, what does that enable? Like what does the world look like when we have more of an open backend API for gaming systems?

[00:26:01] BG: Right. So this is one of the things that we're most excited about with Dark Forest. We've been seeing people building things like third party clients, or automations. Or we have a very rudimentary plug in system, for example, in the game, and a budding sort of community plugins, ecosystem that's getting built out. We're seeing players build out, for example, automations that will automatically ferry resources between planets, and then distributing those plugins. And then those plugins themselves become first class features of a lot of players as games.

So one of the exciting consequences of games being client agnostic is that players can customize the gaming experience to their liking deciding which features do I want to be a part of my core gameplay loop, and which features do I want to do without? If players want, they could re-skin the entire game. At the bottom level, the game is just a series of data transformations. So you can imagine that while Dark Forest, the way that we present it as the devs in our native client, is a space-themed game where you're conquering planets and expanding a space empire. Someone else could come in and make some sort of like medieval-themed skin on top of the game where you're going from castle to castle, or village to village, and dragons are flying around instead of spaceships. So I think that the high-level theme here is that we want to open it up to just see what developers do on top of the game and sort of harness the creativity of a player community rather than dictating from the top down what is the gameplay experience supposed to be like? And what features or mechanics are or aren't going to be a part of the game.

[00:27:30] JM: And I can see why that makes a lot of sense for Dark Forest, because when I look at Dark Forest, I basically see – I mean, I do see the expansiveness of an MMO, like World of Warcraft. You just have like a UI that looks like it's on an Atari machine basically, which is great, which is great. Because who needs – Whatever. You can make the implement of like, really, really fancy UI later on if you want. Like you basically have implemented the command line of Dark Forest.

[00:28:03] BG: Yeah. On the right side of the game, there literally is something that acts as a terminal as well, which you can type JavaScript into. So yeah, just very, very minimal, but functional, hopefully.

[00:28:13] JM: I'd be curious about where your game influences are. This kind of reminds me – I never played it. But I heard of – Have you heard of EVE Online?

[00:28:22] BG: Yeah. Mm-hmm.

[00:28:23] JM: Okay. So I've heard that EVE Online is kind of like this. But what are your inspirations when it comes to the design of the game?

[00:28:32] BG: For sure. So the core mechanic of the game, which is basically just transferring this resource we called energy between planets, is inspired by a class of games that I think descended from a game called Galcon, which was basically this relatively simple space RTS game. You have planets. They have energy on them, or population, or army, or something like that. Over time, the planets are producing more and more troops, I guess. I don't actually remember what the resource was called. But the idea here was that you could just click and drag from one planet to another to transfer troops over. And the idea is you'd get dropped into a map with a bunch of planets on it with either like an AI player, or maybe another human player. And you'd both be trying to conquer this map and destroy the other player by taking over their planet. So you'd want to pick out the planets that had like strategically placed locations or like higher growth rates or stuff like that. So that's kind of like the inspiration behind the core mechanic of the game.

The way I came upon this genre was when I was a kid, I remember being on like armorgames.com and playing a game called Phage Wars, which was essentially like a re-skinning of this, but with like cells. So you're like a bunch of bacteria in a cell culture on a petri dish and you're like spreading out your colony between all these different cells. But beyond that, what we've found is I think that a lot of – Like the blockchain is a pretty constrained execution environment. And so when we decide to try to come up with different mechanics, a lot of times they're pretty much just like the simplest thing that puts us on the board on a certain dimension in terms of having a strategy game. So there's like one economic resource, and that's silver, and that's just mined on a certain planet. It can be moved between planets, and it can be used to upgrade planets. So I think the core of the game really is just that energy moving mechanic.

[00:30:20] JM: Tell me a little bit more about what the player is doing in the game. And then we can talk about the architecture of how you're actually building this.

[00:30:33] BG: Yeah. So if you're a player and you're coming in, and you are not coding, you are using our default web client. And the default web client basically allows you to do a small number of fairly simple actions. So the first one is that energy is a resource that is naturally going to grow over time on planets that you own. You can move energy from a planet that you own to another planet that you own, just this simple kind of click and drag. You can decide what proportion of the energy on the origin planet you want to send. And if you are sending energy to

an unknown planet that doesn't currently have another player owner, sending energy to that planet will conquer it. So that's sort of the most basic like core gameplay loop.

Now, besides planets, there're a couple of other celestial objects in the game, such as asteroid fields, where you can mine silver, which is the economic resource of the game. Now, if you conquer an asteroid field, it will start to build up this economic resource over time. And then you can ferry that over to other different planets and use that to upgrade your planets, upgrading various stats, like the speed at which they can send moves, or the range that they can send moves, or the defense of the planet, which is useful if you're under attacked by another player. And that is pretty much it.

Beyond this, pretty much everything else is going to be constructed by the players. So, for example, you might run into another player. And there might be like some very valuable or important planet that both of you would like to hold. You both might engage in some sort of negotiation. Like I might trade some map data that I have that I've explored, like this section of the universe, you haven't explored it. I might trade that for access to this planet. Yeah, other than that, we pretty much leave it up entirely to the players.

Oh, I guess the last kind of important component that was introduced in the last two versions is this notion of artifacts. So, on certain objects, called foundries, if you conquer a foundry, then you can prospect the foundry to find an NFT artifact. And this is essentially an item that allows you to power up a planet. It's almost like a transferable upgrade. So some artifacts will allow your planet to have a stat boost increase. Others might confer like some sort of special effects, like, for example, the black domain artifact allows you to essentially just nuke a planet, just destroy it and remove it entirely from the game. So there're also these NFT items that are floating around the universe, and players are sort of contesting. And I think that gives rise to a lot of interesting dynamics as well.

[00:32:58] JM: I don't think we can do the game and its interface justice over audio. So I would encourage anybody who is intrigued at this point to check out the interface. It does look like basically like a screen from an Atari game flanked by two text terminals, which is pretty cool. So now that we've given a bit of an outline for what the game is and how it looks, talk about the architecture. Like how is this game actually written and deployed?

[00:33:32] BG: Right. So the thing that corresponds most closely to a traditional game backend is a set of smart contracts that's deployed to an Etherium virtual machine compatible blockchain. So, I think, for the first versions of the game, we were deploying on the Ropsten test network, which is essentially a test net that's going to be more or less identical to the Etherium main net, but it's not really endowed with economic value. Its purpose is mainly for testing out applications in low cost settings. Now we deploy our contracts to xDai, which is a sidechain. Essentially, that's a blockchain that uses a similar sort of virtual machine that Ethereum does. It's the same EVM, but basically, there's a different consensus mechanism. And we treat it as sort of like a staging environment for the game while it's still in beta, because Etherium gas costs are super high, but gas costs on these sidechain staging environments are much, much lower.

So we have this set of smart contracts that's deployed onto the EVM. And, additionally, we also provide a default web client for players to interact with the game with. So if you go to the URLs, zkga.me, zk Game, you'll load up a default web client. It's just a static site that essentially connects to a blockchain node and allows you to download data from the blockchain so that you can build up your representation of the game in your browser. And whenever you make a move, it will send a transaction up to the block blockchain that you'll have to wait for that transaction to get confirmed and mined and then it's reflected in your browser.

Additionally, beyond that, we also provide – The game is open source. So we also just provide all the client code online on GitHub as well. And I know that a lot of players basically like will fork this client repository, customize it to their own liking, and basically run that locally. And they're able to connect to the blockchain as well either by connecting to a public blockchain node or by running their own blockchain node.

[00:35:28] JM: As you described, the application runs on the xDai network, this Ethereum compatible network that is not on Etherium so you can avoid the gas costs. Out of curiosity, how does that network work? Is it just like you've only got like there's a few boxes out there? Are they on AWS or something? Like what is the xDai network?

[00:35:54] BG: Yeah. So the xDai network is essentially going to be – It's going to be very similar to the Etherium main net, except with a different consensus mechanism. So on the

Etherium main net, Ethereum uses a consensus mechanism very similar to Bitcoins, which is proof of work. So at all times, you have all of these mining nodes all over the world basically racing to try to put together the next block and mined out some hash pre-image that corresponds to a hash starting with some number of zeros. And that consensus mechanism is a very like open and decentralized and permissionless one. Anybody can join it. And, obviously, there're downsides as well in terms of the energy costs and things like that, but that's what Etherium has been on for many years.

On the other hand side, some side chains like xDai exist, which basically are closer to a hybrid like proof of stake or proof of authority sort of consensus mechanism. So in particular, on xDai, I believe that there are about a dozen validators that these are like the dozen special nodes that are permissioned to mine blocks and push them up to the network. These validators can vote on various protocol governance things, like adding validators, or slashing other validators. If validators act not in accordance with the protocol, i.e. like they do something illegal, then they can be slashed with the stake token, I believe. And, yeah, it's not going to rank as high on decentralization. But it is very convenient, I think, as a network with low gas fees and a place to kind of prove concept for a lot of applications. And there is a lot of real financial activity, I think, being transacted on the xDai network as well.

[00:37:33] JM: So this is fascinating to me. So xDai is actually – There's actually people working on it. And it's like a profitable sort of thing, because they've got a token.

[00:37:42] BG: Yeah. So I believe that the – and I'm not 100% sure if this is exactly technically correct. But one of the kind of foundational parts of the xDai network is the stake token. And what's going to happen is that each of these about like a dozen validators are going to put up some stake token and, essentially, the protocol is going to say that if one of the validators acts in a way that's not in accordance with the protocol rules, their stake token can be burned, or slashed, or probably like redistributed to the other validators, or something like that. I'm not 100% sure, but I'm aware that like at least like there are a class of sidechains that have mechanisms such as this.

[00:38:18] JM: Yeah, it's interesting to me, because I hadn't even thought about – I guess, I mean, the use case that you're basically describing is staging infrastructure for Etherium smart

contracts, or I guess more centralized infrastructure. It could be used for staging. I'm not super familiar with the crypto ecosystem. So this is an interesting realization for me that there is this kind of staging infrastructure – Or infrastructure that can be used for staging.

[00:38:48] BG: Yeah, and this is kind of what test nets also were are used for largely. So, for example, Ethereum has a number of test nets, many of which have different consensus mechanisms. The Ropsten test net uses the proof of work consensus mechanism. So you can test your miners on that one. Other ones are proof of authority. So, basically, there's like a small permission set of nodes that can push new blocks to the network. And oftentimes, before a production release on main net, applications will test out deployments and stuff in these kinds of test environments.

[00:39:17] JM: Right. And maybe this is a naive question. But given that you are in staging, or you're thinking yourself in staging, why isn't there just some way to like run this on like a few AWS nodes? Like why would you just do that?

[00:39:33] BG: Yeah. So I think that the eventual goal is definitely to get something that is deployed on a completely decentralized and permissionless blockchain. But in order to do that, we have to build with that in mind now. So that means that we have to build our smart contracts using the solidity programming language, or at least like some programming language with a compiler that can target the Etherium virtual machine. And I think that we could, for example, do things like spin up our own proof of authority network with nodes that are run on like AWS boxes or something like that. But for now, if our eventual goal is to target some sort of like layer two or some like EVM compatible chain, or construction, then, for now, we've just found that it's been very, very easy and convenient to deploy onto xDai. And it would actually take us, I think, more effort if we wanted to spin up some AWS boxes, run our own network, or even run something that like mocks the blockchain.

Early on, we actually did build out what we call the mock chain, which was a web server that exposed the same interfaces as a blockchain would, at least to our web client. And we were able to sort of like write the logic of the game in JavaScript and stuff like that. But that actually turned out to be like more of a hassle, because then we were maintaining like the mock chain as well as the smart contracts, which we're eventually going to get deployed to a real network and

this like client that had to be able to connect to both. So, yeah, I think it is like a sufficiently different environment that it's been easiest for us to develop specifically for the blockchain.

[00:41:09] JM: Alright. So continuing our crypto gaming bingo board, I have to ask you about issuing an in-game currency. Can you do that in a legal way? I'm very curious about the legality of in-game currencies and how you can use them in-game.

[00:41:32] BG: Yeah. So we actually have not been thinking about this question too hard, because for us right now, like the goal is really just to build a fun gameplay experience first. So, for example, like there aren't really economic mechanisms built into the game, which I think is actually – Like, from my perspective, the thing that excites me most about crypto is the tech and this idea that these applications are fully composable, client agnostic, interoperable. They can lean into a lot of – Like they can harness a lot of third party developer ecosystem energy.

So, for us, we actually – Like we don't currently have any sort of concrete plans to issue in-game currencies or something like that. Like, who knows, in the future, it may be possible? But we've found that by focusing on the tech and on the gameplay experience, it's had this nice second order effect of the community that's starting to get built around the game is very focused around stuff like development and the game for intrinsic reasons, rather than extrinsic reasons, like is the token of the game going to go up, or something like that? We're trying to avoid building out like a community that's based on like speculation or something like that. At least for these like early days, we really want to focus on making a good core gameplay experience.

[00:42:40] JM: Certainly, but think about – I haven't played any of these games in detail. But as I understand, Second Life, World of Warcraft, Diablo, these games have currency in them, and the currency creates this really fun, second order ecosystem within the game. And like crypto seems to just supercharge that, right?

[00:43:02] BG: Yeah, that's definitely true. And we do know that like some economic activity is happening out of band. Like players are trading items for – Or like artifacts in the game for other cryptocurrencies. Or they're making trades of planets or various other assets within the game. So we think that stuff is going to evolve naturally. In fact, like because crypto makes it so frictionless for this to happen, it's got to happen at some point. But I think, like for us, one of the

advantages of being on a blockchain is that we do not control this activity. For example, we're not controlling the on and off ramps where people are loading money or value like into their cryptocurrency wallet. That's something that's happening on a lower level of abstraction from us. And because of that, I think that helps alleviate some of the concerns about like, “Are we facilitating the transfer of these certain assets with economic value?”

I know that like a lot of games are really cautious about like out of band markets, where say like RuneScape GP, or WoW gold, or something is being exchanged for money for a variety of reasons. For us, like the interesting thing about building on a blockchain is like, just like the game is client agnostic from the very beginning, like we simply cannot control what players are using to play the game. Similarly, any sort of economic activity that players are engaging in is something that is, to begin with, out of our control. We're deploying these smart contracts up to the blockchain. Anyone can interact with them however they like. In some sense, we are just one of many participants in this open and permissionless ecosystem, and no participant – So long as there aren't – Once like things like admin controls and stuff are removed from the contract, no participant can dictate what economic activity does or doesn't happen.

[00:44:44] JM: Of all the things you could work on in crypto, there are so many domains you could focus on. Why are you focused on gaming?

[00:44:54] BG: I'm personally excited about gaming, because I think that gaming allows us to present a vision for the future in a way that might not be realizable with more, say like, “high stakes applications” today. So, for example, I think that a lot of interesting activity is going on around the idea of interoperable contracts built on Dark Forest that might allow for things like player guilds or organizations. And I think that in the long run, these might allow for a safe experimentation ground for things like DAos, or governance structures, or other things for crypto native worlds.

So, to me, it's really exciting, because we get to prove out a lot of things that could one day become core patterns, whether in blockchain technology and programming, or blockchain governance and organization, but in a way that we can we can iterate very, very quickly. Obviously, it's also very fun to build a game. Like I played a lot of games as a kid, and in the

day-to-day, it's just exciting to try to put together a compelling set of mechanics as well. So that I think is sort of the bundle of resources that gets me really excited about crypto gaming.

[00:45:59] JM: So you see Dark Forest as essentially like a testing metaverse for ideas in a post-crypto society?

[00:46:14] BG: I think so. I think at least it's a safe place for experimentation. I think, for example, like you use the word metaverse, which I really, really love. And I think that, for example, the metaverse of the future, the shared digital space that we all interact on, is not going to be top down designed. It's not going to be like a universe that some company decides to like write out a product roadmap for and then do like a round of internal testing, and then they release it to the world. And then, "Boom!" Like a couple years later everybody is on the metaverse. And that's also not the metaverse that I would want to see in the world either. Like I want to see something that's grassroots, built from bottom up experimentation. Perhaps it's the combination of a bunch of open protocols and standards. And I think that Dark Forest hopefully can be a part of whatever that grassroots process is of experimentation around everything, from governance, to digital physics. Like how do we move around in the metaverse? To the patterns of what are the best leveraged ways to use these blockchain kind of affordances and financial mechanisms. So, yeah, some sort of like proto metaverse grand social experiment. I don't even know that we necessarily have a word for it ourselves.

[00:47:26] JM: Okay, that's really exciting. Last question. Tell me about the canonical engineering problems of building Dark Forest. Like what are the things that you're running into time and time again that are frustrating you?

[00:47:45] BG: Yeah, one thing that we kept running into early on before I had a better grasp of EVM internals was – So you're in a really constrained execution environment. And like one issue is that smart contracts have a limited size. So you can only deploy contracts of up to a certain size up to the blockchain. So that's kind of forced us to try to figure out how to break up logic into a bunch of different like library contracts and understand how you can have like this deployment of a bundle of contracts talking to each other in the correct ways so that we're not deploying any single contract that is too large. So we're sort of like sharding the game logic into

multiple different contracts, which has been like an interesting problem and a fun learning opportunity.

Another thing that I think has been challenging has been – So, for example, like one common pattern that I think more and more applications are using in blockchain is the idea of contract upgradability. So, when you push a smart contract up to the blockchain, naively, you can't easily change the code of what is running. And that's sort of the guarantee of immutability that blockchain provides is both a strength and a weakness. But you can do kind of clever tricks such as routing transactions that players are making through like a proxy smart contract that talks to one of N different implementations smart contracts. And if you discover a bug, then you can sort of swap out which logic contract is being used to essentially like handle that transaction that's just been routed through the proxy contract.

So setting up the upgradable contracts architecture was also kind of an interesting problem as well. And that's been really useful for resolving. Like sometimes a player will come to us saying like, “Hey, like, I noticed this reentrancy error, for example, in how artifacts are harvested.” That was a really fun one to look at. And this, again, goes back to like games are a safe space for experimentation, I think relative to a lot of financial applications. So yeah, the upgradability has been an interesting question.

And I think the last one that's been a perennial struggle, and which we have just now started to get a handle on is I think a lot of people talk about issues with blockchain scalability in terms of transaction throughput. But another problem that we've run into is scalability in terms of reads from the blockchain. So we've been working with the xDai network because players – There's like so much data in this contract. There's like literally like hundreds of thousands of planets that are essentially being stored all on-chain. Players are downloading huge segments of that every time they log into the game, and they're listening for updates to this like massive amount of states. We need to provide players with a default blockchain node that they can connect to, so that way they can fetch all of this data without having to run their own node at the start. Though, I think a lot of more advanced players will be running their own personal nodes. And we've basically – xDai now has been doing a lot of work to maintain like a fleets of these public node endpoints that is sort of hidden behind a load balancer and is able to appropriately handle those

requests. And from our end, it's kind of tuning the way that we fetch data from the blockchain to be more efficient as well.

[00:50:49] JM: Alright. So you've given me basically two requests for startups. The first one is we need continuous integration pipelines for smart contracts. That's the first one. And then we need Redis for the xDai network.

[00:51:06] BG: Yeah. Right. Right. And I think that a lot of these like tools are getting built out in responses to problems like this. So there're a bunch of different companies or protocols that are taking different approaches. For example, one thing we've recently started experimenting with is a tool called The Graph that is a layer on top of the blockchain that will like listen for and index blockchain data.

[00:51:29] JM: Yeah, that's one of the cooler like infrastructure blockchain things that I've seen.

[00:51:34] BG: Absolutely. Yeah, just super useful. Because a lot of times like storing data on the blockchain, you are very much concerned with storing it in the most efficient manner as far as like the amount of data that you're storing. But that may trade off sometimes with things like you won't have well-indexed data that's easy to query. And I think that The Graph does a great job of helping to solve that problem.

[00:51:56] JM: And The Graph uses GraphQL, right?

[00:51:59] BG: Yeah. So what they do is – Like at the consumer level, like the consumer basically gets to just hit a GraphQL endpoint that is going to have –

[00:52:08] JM: They’re like the ultimate GraphQL server basically.

[00:52:10] BG: Yes. So The Graph is super cool. And they're working on doing this all in a distributed and decentralized way as well, which is I personally admire as a proponent of the ethos of the space.

[00:52:23] JM: Okay. Well, Brian, such a pleasure talking to you. Really interesting project. Really insightful.

[00:52:28] BG: Likewise. Yeah, thanks so much for having me. And let me know if you'd like any Dark Forest invite keys as well to give out to any fans.

[00:52:34] JM: Okay.

[END]