Talking About Talking About Web3

Tim Roughgarden

April 1, 2022

Preamble

- thanks for invitation, honor, students at blockchain club are amazing
- in general, student-driven involvement in space has been amazing, I've never see anything like it in tech
- let's talk about web3. actually, let's talk about talking about web3. I've never seen a more misunderstood set of technologies in my entire career (by everyone, including e.g. CS academics).

A Misunderstood Technology

- But really, can you really blame anyone for being confused? I mean, what's this technology even called? Distributed ledgers? Blockchains? Web3? Crypto? Wait, is "crypto" short for "cryptocurrencies" or for "cryptography"? (For this talk, let's stick with web3, referring broadly to the tech stack, including both blockchains and the application ecosystem built on top of them.)
- As you can imagine, and like many of you, these days I field lots of confused questions from friends, colleagues, family members etc. who want to talk to me about web3. In fact, it's worth enumerating all the totally orthogonal dimensions along which many smart people are simultaneously confused.
- A typical sequence of questions might go: (i) what's the different between PoW and PoS? (ii) why would anyone pay anything for a jpeg? (iii) which should I buy, BTC or ETH? And generally, I don't think the questioner realizes that these are three radically different questions.
- For an analogy, imagine we were instead talking about the Internet, and someone asked, in quick succession: (i) how does the BGP protocol decide which routes data packets travel along? (ii) why are teenagers so obsessed with TikTok?; and (iii) which tech stocks should I buy? Here, I think most people would recognize this as a schizophrenic

conversation, as these questions are careening wildly between basically unrelated topics: (i) is about how the technology works, and is something you'd ask an engineer or computer scientist such as myself; (ii) is about how people use the technology and their choices and behavior, and thus best addressed to those in the social sciences (economics, psychology, etc.); (iii) is obviously about investing, and a question you'd pose to your financial advisor.

How It Works vs. What It's Good For

- My goal in this talk is to suggest a mental model for, at the highest level (i.e., least level of detail), organizing discussion around web3. A lot of what I say may be obvious or trivial to many of you. But, in my experience, it's usually worth taking the time to say the obvious parts out loud.

- The first point is to always be clear in your mind whether your talking about how web3 technology works (i.e., below the user-facing application layer) versus how it might be used (the user-facing application layer and stuff built on top).

- E.g., PoW vs. PoS concerns implementation of the technology, NFT marketplaces like OpenSea are examples of how it might be used.

An Abstraction of Web3's Computing Functionality

- Computer science prides itself on abstractions—models of computing functionality that, to paraphrase Einstein, are as simple as possible, but no simpler. Central to the high-level mental model, and sitting between the two sides (implementation and applications), is such an abstraction, an idealized model of the computing functionality offered by Web3.

- Thus, the three parts of the diagram correspond to the questions: (i) what is the core computing functionality that Web3 offers?; (ii) is it possible to actually realize that functionality with current technology, and if so, how?; (iii) were that functionality to actually be implemented, what could we do with it?

- The abstraction (i) sitting in the middle usefully decouples thinking about implementation from thinking about applications. You can discuss hard engineering challenges in isolation, with the goal of implementing the desired computing functionality (i) rather than any application per se (iii). And you can discuss potential applications of the technology (iii) before the technology even exists (ii) (hypothesized on the eventual implementation of the functionality in (i)).

- For example, in quantum computing, theoretical computer scientists came up first with appropriate abstractions of quantum computers in the early 1990s (answering question (i)), then quickly came up with interesting applications of quantum computers should they ever be built, like Shor's factoring algorithm (answering (iii), predicated on eventually answering (ii)), and these days many research groups are working hard to actually realize quantum computers (i.e., answer question (ii)) that support the functionality promised by the abstraction in (i).

- This may all still seem a little, uh, abstract, so let's go through these three questions in a familiar and relatively well-understood example, the Internet.

Comparison: The Internet

- Let's start with question (i). What is the core computing functionality offered by the Internet? Let me suggest: the permissionless and near-instant transmission of data between any two points on the globe—like the Post Office, but way faster (good news) and digital only (both good and bad news). These days, almost anyone would recognize this (or something like it) as a description of what, fundamentally, the Internet does.
- Question (ii) would involve explaining how e.g. the IP protocol works, maybe DNS, etc. You would learn all this in a standard engineering course on computer networking. 99% of Internet users have no idea how the Internet works. And that's a great thing! Given that so few people in the world are hardcore techies and nerds, a logical consequence of the widespread adoption of any technology is that almost none of its users know how it works. This is what we want to aspire to!
- Speaking in 2022, there are a zillion obvious answers to question (iii), that anyone would be able to tell you: YouTube, TikTok, social networks, online dating, search engines, Wikipedia, etc. But let me tell you, as someone who was using the Internet 25 years ago, in 1997 none of those applications existed. All you could do then was send email, transfer files, and (as of very recently) read static Web pages. And 1997 was 27 years after the Internet debuted (in the form of the ARPAnet), mind you—whereas it's now only about 13 years after Satoshi Nakamoto first dropped the Bitcoin protocol.

Reflections

- In the case of the Internet, almost anyone can answer question (iii)—this is basically the definition of mainstream adoption.
- But interestingly, most people have a decent sense of the Internet's fundamental functionality and could offer a reasonable answer to question (i) (or at least would recognize the answer if it were presented to them).
- Question (ii) is the esoteric one that almost no one knows the answer to. But who cares? As long as there's a small group of nerds that figured out how to pull it off, we should be good to go.

- If Web3 goes truly mainstream, we should expect to see the same pattern: everyone will be able answer (iii), many people will have a decent sense of (i), and understanding (ii) will be a niche (but valuable!) skill possessed by only a select few.

The State of the State

- So, mainstream adoption means: everyone talking about Why, decent understanding of What, blissful ignorance about How.
- What does the discourse look like now in Web3? Totally different! (It's still early. . .)
- Majority of discussions, even in mainstream (non-technical) media, are about How (as opposed to target of perhaps 1%). E.g. the word "blockchain" really only concerns the How (what are the blocks? what's the chain?) and not the What or the Why (which are about the abstraction and applications, not the implementation/data structure details). ("Web3" sounds a bit gimmicky to an academic's ears, but at least it emphasizes the full technology stack as opposed to just the plumbing.)
- I sympathize! There are many, many really cool things to say about the How (my Columbia courses are almost entirely about the How). But remember that, ultimately, there should be no one other than engineers and computer scientists worrying or caring about any of the implementation details. (My courses are specifically for CS students, that's my excuse. Eventually I'll roll out courses aimed at a wider audience and accordingly focus more on the Why and less on the How.)
- In addition, we probably don't yet know the eventual answer to the Why question—I suspect that what will in hindsight be viewed as Web3's killer apps haven't been invented yet (just as the killer apps for the Internet hadn't yet been invented by 1997). And I suspect that the right people to invent them are exactly the people here in this room—young, brilliant, and hungry for a new generation of the Internet and the Web.
- We do at this point have proofs of concept that the computing functionality offered by Web3 enables fundamentally new applications. Parts of decentralized finance (DeFi) and the NFT space demonstrate this, for example. So while I think it's reasonable for someone to express skepticism about whether Web3 will ever have any truly game-changing killer apps, personally I'm not concerned—history shows that whenever we implement a fundamentally new type of computing functionality, people figure out amazing things to do with it. I see no reason why that pattern wouldn't continue to hold for Web3.
- But, as an academic computer scientist, what really bothers me is the lack of informed discussion of the What. What, fundamentally, is the functionality offered by the Web3 stack? Only by answering this question can we speak clearly about the goal of all the engineering work that's going on, and about how this new functionality might usher in a new generation of applications.

- So, let me give you my answer as to the "What" of Web3.

The What of Web3

- I believe that the right mental model for the idealized functionality offered by the Web3 stack is: (note: not saying this is literally true, just that this is a good way to think about it)

  - a general-purpose computer (like your laptop, or rather a virtual simulation of your laptop); (but wait, we already have computers. and today's web3 computer is about as powerful as the vacuum-tube computers of the early 1950s)
  - no owner or operator (what people mean by "decentralized") (in effect, just running on its own in the sky as a public good) (obviously very different from "computers" as we usually think of them—e.g., turning off the web3 computer basically as hard as turning off the Internet)
  - anyone can use it (perhaps subject to usage fee, if bumping up against capacity constraints, but no access list, no permission or credentials necessary), either to deploy new applications ("smart contracts") or take advantage of existing ones;
  - supports user-owned data. (Think digital property rights. You can prove without question that you own something, and no one else can fake such a proof. Transfer of ownership is possible, but only with the permission of the current owner. The computer tracks current owner of each piece of data.) (So no owner of the computer, but there can be owners of specific pieces of data stored on that computer. What Satoshi Nakamoto basically figured out was how to get these seemingly contradictory properties at the same time. Satoshi focused on tracking ownership of Bitcoins specifically, but the solution applies more generally to arbitrary digital data (such as NFTs), not just digital coins.)

- other properties as well (e.g. transparency, computer is effectively always emitting a ticker tape with every instruction its ever carried out) but I suggest focusing on the above three

- each of these points is very important. if your only takeaway from this talk is that the functionality exported by web3 is like that of a computer, i'll consider it a victory. in particular, *never call it a database!* i have no idea why the "blockchain as DB" meme has persisted for so long. the point of a database is merely to store data and support efficient queries over that data, not to execute arbitrary computations. modern smart contract platforms (Ethereum, Solana, etc.) are "Turing-complete," meaning that they do support arbitrary computations (at least, those of at most a certain length) and are analogous to computers, not databases. (even Bitcoin would more accurately be compared to a restricted (non-Turing-complete) computer than to a database.)

- note not at all a new observation: back in 2015 when Ethereum launched, the founding team referred to it as the "world computer." has not caught on as much as it should have. ("web3" goes all the way back to 2014 and is only now getting traction, though, so maybe "world computer" will get its chance in the limelight well.)
- the other points are crucial because, if we are ever to have a satisfying answer to the Why question (i.e., applications with truly mainstream adoption), it will be because this combination of properties (which is not possessed by traditional computers) enable valuable and fundamentally new applications.
- again, this is the main part of the talk: in your mind, think of the Web3 stack as a computer that lives in the sky, executing on its own, usable by anyone, and enforcing property rights for user-owned digital data.

A Little Bit About the Why

- I said earlier that asking about Web3 killer apps in 2022 is like asking about Internet killer apps back in 1997. It's easy to imagine that the answer to that question will be much more obvious and compelling in ten years than it is now. But, given that all of us are going to get asked about killer apps at our next Thanksgiving dinner, how should we answer this question now?
- There are a number of reasonable answers (e.g., you can blow away finance nerds by telling them about how flash loans exploit the atomicity of Ethereum transactions), but let me just single out one that's been working well for me lately. It's less about a specific application that about the inevitable collision between two undeniable forces—I want to give my conversation partner a feeling of inevitability, rather than point them to any one URL. (Though feel free to think about Bored Apes, for a concrete example).
- The first undeniable force—a timeless one—is that humans often define their identity in part through what they own. Why do people pay so much for a Rolex, a Ferrari, or a Chanel handbag? Are those products really so superior to the next-best option? No. Maybe it's self-expression; maybe it's peer pressure; maybe it's simply conspicuous consumption. But whatever the reason, some people apparently care a lot about the non-physical properties of these goods (like the brand name).
- The second undeniable force, which is a generational one, is that people's identities increasingly exist in the digital world (e.g., in one's Instagram or Twitter feed). In the current Web2 world, individuals' digital identities are largely controlled by big tech companies, not by the individual themselves (e.g., if Twitter wanted, it could delete my account and all my content would be lost forever).
- If you accept these two forces as undeniable, then it sure seems like there will be serious demand for digital goods (that express identity, among other things) with traditional-looking property rights (e.g., verifiable ownership, permission to transfer, prohibition of

seizure). And looking at the Web3 abstraction (a computer with no owner or operator that supports user-owned data), it sure seems perfect for satisfying that demand. (And, we're already starting to see it happen, perhaps most obviously with PFP NFTs that live on a blockchain such as Ethereum.)

Web3 and Cryptocurrencies

- I've spent this talk harping on one of the main reasons so many people seem so confused about web3: little and misguided discourse as to the What of Web3 (the fundamental computing abstraction offered) and, relatedly, the lack of separation in discourse between the How (how it works) and the Why (what's it good for?).

- But there's another extremely powerful force sowing confusion, which is the conflation of web3 with cryptocurrencies.

- tl;dr: from a modern (post-Ethereum) viewpoint, cryptocurrencies are the means, not the ends. The goal is to achieve the functionality of a programmable computer that lives in the sky; imbuing the implementation of that functionality with a native currency just so happens to be a very useful tool (to charge for usage, reward actors that contribute to that functionality, proof-of-stake sybil-resistance, etc.)

- You may have seen talks or writings by smart and serious people, often backed by compelling points, evidence, and analogies, about why blockchains are doomed to fail. In most cases, what is actually being argued is that the *cryptocurrencies* hosted by blockchains are doomed to fail as serious alternatives to traditional fiat currencies like the US dollar (due e.g. to volatility). Maybe they're right, or maybe not, I'm not sure. But I also don't really care—even if cryptocurrencies never seriously compete with traditional currencies, they can (and will, I think) succeed as an implementation detail under the hood (potentially never seen by end users) of game-changing applications enabled by web3 functionality. That is, I expect them to at least succeed as the means, if not as the ends.

- Web3's seemingly unbreakable association with cryptocurrencies is a testament to the power of first impressions. Satoshi Nakamoto was quite clear that the primary goal of the Bitcoin protocol was to create and manage a digital currency that was beyond the control of nation-states. That is, for Nakamoto, cryptocurrency really was the ends, not the means to any other type of functionality. The overwhelming majority of all discourse around Web3 technology prior to 2017 was about Bitcoin, resulting in widespread and hardened beliefs (persisting to this day) that blockchains and cryptocurrencies are really the same thing. But now that we're in an era of general-purpose smart contract platforms (Ethereum, Solana, etc.), such beliefs are badly out of date and urgently need to be updated.

- To be clear, I'm not dismissing Bitcoin at all. In my opinion (which is also now the mainstream opinion), Bitcoin has become startlingly successful as a store of value. For example, not many people would tell you that it's obviously better to park some money in gold than in Bitcoin. So while Bitcoin may never become the payment of choice for a cup of coffee, it nevertheless manage to create an entirely new asset class, effectively out of thin air. This is an incredible feat! One that I don't necessarily expect to see again (at least at this scale) in my lifetime.

Conclusions

- think of the Web3 stack as implementing the functionality of a general-purpose computer (not a database!)
- what's special about this computer is that is has no owner and yet enforces property rights for user-owned digital data
- the details of how this functionality is implemented is super-interesting computer science, but for mainstream adoption it's important that 99% of Web3 users have no idea how it works
- the killer apps of this functionality will become clearer in the coming years, but the inevitable collision of identity-driven ownership and digital identity point to a huge demand for digitial goods with traditional property rights
- web3 is not the same as cryptocurrencies, and can take over the world even if no cryptocurrency ever becomes a viable alternative to existing fiat currencies.