CHAPTER THREE

Trustlessness

CRYPTO CONSENSUS VIEW

Traditional businesses are trust-based while blockchain applications can be trust-minimized.

UNBOUNDED CAPITAL VIEW

Current applications of blockchain technology don't minimize trust but instead shift trust from traditional counterparties onto code and developers.

In the absence of efficiency, trustlessness is one of the major qualities of Bitcoin and other blockchains that inform the crypto consensus view of what makes these technologies valuable. The crypto consensus imagines that reliance on trust is something that blockchain can be used to minimize or eliminate. This ability to remove trust is thought to be a major source of blockchain's value relative to traditional options. **The Bitcoin wiki** states that "Bitcoin is only useful if it is decentralized because centralization requires trust. Bitcoin's value proposition is trustlessness."

Blockchain applications are often referred to as trustless or trust-minimized. While the crypto consensus acknowledges that trust has served an important and useful function in the world to this point, its necessity poses a threat that many would like to avoid. The theory follows that as trust-minimized applications become more and more efficient,

users will increasingly opt to eliminate the need for trust rather than continue to rely on it and risk occasionally experiencing severe consequences from doing so.

In our view, trustlessness is a misnomer. Rather than being trustless, these applications place an extreme level of trust in code and the developers who create that code. The results of this effort are less trustworthy applications. We believe that applications and blockchains seeking to promote trustlessness at the expense of efficiency are highly unlikely to be successful since they are pursuing a goal with little to no value over a goal with immense value.

WHERE DOES TRUST-MINIMIZATION COME FROM?

It isn't surprising that a narrative formed about how blockchains can be used to minimize or eliminate trust when one considers that the introduction of the **Bitcoin whitepaper** is a description of the issues that stem from needing trusted third parties in internet commerce, an issue Bitcoin was designed to solve. However, reading precisely what Satoshi wrote in the whitepaper is extremely revealing and informative about the nature of the problem Bitcoin solved.

> *Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.*
>
> *What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could*

easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

Our interpretation of this section is that Bitcoin seeks to be able to remove trusted third parties from internet commerce for a very specific goal: micro-payments. Imagine a version of Google that doesn't know who you are. You make a search, you attach a micro-payment, and you get results. Nothing is tracked. This is impossible without non-reversible micro-payments. The need for micro-payments is obvious, but these also need to be non-reversible or else users could just revoke the money even though Google can't revoke the search. By eliminating the need for transaction processors with a legal obligation to mediate disputes, these casual, non-reversible micropayments become possible.

We do not see this section as evidence that Bitcoin or its blockchain technology can remove the general need to trust third parties. We also don't see anything here that suggests that third parties aren't trustworthy or even generally helpful. It simply points out a specific application that is prohibited by a specific inefficiency of internet payment intermediaries and provides a solution. If anything, the whitepaper makes it seem that the customers are the ones who are untrustworthy. It doesn't put any weight behind a general idea that financial institutions are not trustworthy from the standpoint of providing services.

THE CRYPTO CONSENSUS CONFUSES TRUST WITH ECONOMIC INCENTIVES

The crypto consensus, going far beyond the contentions of the whitepaper, makes a much larger claim about the role blockchains can play in eliminating or minimizing trust. Consider this passage from Multicoin Capital's **Crypto Mega Theses**. In describing what unites their three crypto mega theses, Open Finance, Web3, and Globalized State-Free Money, they note:

The common theme underlying these theses is reducing trust between transacting parties. The modern economy is built on compounding layers of trust. We trust tech giants, banks, insurance companies, the government, and more every minute of every day.

We trust so many institutions that we take for granted just how many layers of trust the economy is built on. When we're born and raised with certain trust assumptions, we don't even recognize them as assumptions anymore. Given global complexity, detecting abuses of trust is more difficult than ever before (e.g. Facebook + Cambridge Analytica, Marriott/Target hacks, Equifax hack, etc.).

For the first time in human history, using open networks bound by cryptography and free-market economics, we can incentivize specific human behaviors without creating new trust assumptions. This is a subtle but profound shift.

This is not to say that trust is intrinsically a bad thing. However, all risk is built on trust. By creating a world with fewer trust assumptions, we can reduce systemic risk, and create ultimately healthier and more productive economies and societies.

They go on to note that "Trust is the foundation on which all financial services are built."

In our view, they have misidentified trust as the foundation on which so much of society is built. At Unbounded Capital, we see economic incentives as the foundation of trust, and thus the foundation of whatever trust is built upon. In starting at the trust layer and not the deeper economic incentives, the crypto consensus creates, perhaps unintentionally, a deceptive narrative. In many contexts, the ability to trust someone is rare, special, and sacred. You may trust your bank, but probably not in the same way you trust your best friend. By using the word trust, which carries this emotional context, it is easy to make the goal of eliminating the need to trust so liberally seem worthwhile.

Ultimately, the crypto consensus succumbs to its own narrative and fails to see that the blockchains and applications they promote do not actually eliminate or minimize trust. Instead, these technologies shift the burden of trust from individuals and businesses to code and developers. In advocating for their specific form of trust minimization, they are advocating for a shift from trusting the economic incentives of traditional counterparties to trusting code and the economic incentives of the developers writing that code. This may seem like an appealing shift, even if trust isn't really minimized, but we will show why in practice one should expect this shift to actually make applications less trustworthy.

TRUST STEMS FROM ECONOMIC INCENTIVES

Oxford defines trust as the "firm belief in the reliability, truth, ability, or strength of someone or something." In the world of economic transactions, reliability and truth take

outsized importance. When choosing a bank, one typically would want the services the bank claims to offer to be delivered reliably and for the information the bank serves them to be true. The same applies to blockchain and blockchain apps. These need to function reliably and the information on these blockchains needs to be true. In this case, truth means that information on the blockchain is what it should be based on the inputs and the rules of the system.

Without considering the economic incentives in the world, the source of so much trust would be utterly inexplicable. Why do all of the things we trust tend to work so wonderfully and consistently? The bank functions how it is supposed to because it wants your business. Computers reliably output the right number because if they didn't, they would be worthless. An entire system of law has been developed over centuries to create economic incentives which promote more cooperative, civil behavior. Individuals' actions are thoroughly dominated by their desire to gain economic benefit while avoiding economic loss. These incentives are not always expressed in dollars or some other currency, but they still exist.

As Multicoin Capital points out, performing calculations about the incentives of one's counterparties to evaluate trustworthiness have become second nature for many. Consider that most would be happy to buy a sandwich from a cafe but would not accept a free sandwich from someone standing out on the street giving them away. The cafe has an incentive to provide good service to get repeat customers. Further, if they did something like serve spoiled or poisoned food, they are easy to track down and hold accountable. The person on the street doesn't have the incentive to give a good product and is much harder to hold accountable. The risk that this is a deranged person handing out poisoned sandwiches is too high for most people to accept the offer, especially since other motives to give away sandwiches are less clear.

TRUST-MINIMIZED APPS MAXIMIZE TRUST IN DEVELOPERS AND CODE

Even if economic incentives are sufficient to generate trust, these blockchains would be extremely valuable if they could provide a genuinely trustless solution. Unfortunately, these solutions are not actually trustless, but instead shift the burden of trust from traditional entities to autonomous code. Even if this shift sounds good on the surface, such a distinction between traditional entities and autonomous code cannot actually be drawn.

Instead, the shift is from one set of individuals and businesses onto another, one that often has far less trustworthy economic incentives than the traditional counterparties.

Computers are very trustworthy. They are used to compute numbers and they are exceptionally fast and reliable at doing so. They can also store a vast amount of information reliably in small physical spaces. The strength of computers at generating true, reliable computation and storing information has changed the world in a dramatic way. Many tasks once delegated to people are now delegated to computers, which are much more trustworthy when it comes to computation and storing information.

The trust we extend to businesses promising to deliver services has become almost universally intertwined with the trustworthiness of computers. Hire a bank, and you are hiring the bank's computers. Companies have an incentive to use computers wherever those computers provide a cost-effective advantage. They also have an incentive to refrain from using computers if computers are not cost effective or if computers threaten the trustworthiness of a product. Choosing to trust a bank versus a trust-minimized open finance protocol is not a choice between trusting people and trusting computers. In both cases, it is a combination.

In the crypto consensus, computers' superiority in computation is extended to mean greater universal trustworthiness of computers relative to humans, even if it isn't always articulated in this manner. Consider this excerpt from Multicoin Capital's **Crypto Mega Theses**.

> *The key innovation enabling open finance is the modularization of financial primitives. By modularizing financial primitives, the open finance stack commoditizes trust such that no application has a unique trust advantage over any other.*
>
> *Modularizing financial primitives is an abstract concept. What exactly does it mean to modularize financial primitives?*
>
> *Over the last 24 months, a number of open finance protocols have launched. All of these protocols are modular, and are being used by higher-level applications (and often combined). None of these protocols market to end-customers, provide customer service, or deal with local laws. These protocols are just pieces of code that live on blockchains. This is comparable to how email is built on a suite of open protocols like SMTP, TCP/IP, and HTML/JS to render email in the browser.*
>
> *For example, let's consider BlitzPredict (BP). BP is an exchange focused on sports betting built on top of the Augur, 0x, and (in the near future) Maker protocols. BP relies*

on the Augur protocol as a means to create different kinds of markets, create shares in those outcomes, and ultimately resolve markets. BP relies on the 0x protocol to trade shares between users. And BP will soon rely on the Maker protocol for its collateralized stablecoin, DAI, to denominate trades. Each of these protocols function independently. Because they are modular, a higher-level application like BP can combine the underlying financial primitives to produce a trust-minimized user experience that was never before possible.

It isn't clear how the conclusion, "No application has a unique trust advantage over any other," is drawn. Perhaps one could conclude that each of the applications had no trust advantage if these applications are combinations of various on-chain code that themselves are all equally trustworthy. Since these applications are trust-minimized, presumably the code is trust-minimized, or fully trustworthy. If they were not, then applications combining them would have variance in their trustworthiness.

Computers are very trustworthy, but code is unfortunately far less trustworthy. That is because code comes from humans, and instructing computers is a complicated task for humans. The complicated nature of this task is what has made the ability to write code such a lucrative skill. Further, actually predicting how code will function has a cost. Intentions are one thing, but any developer knows that code doesn't always work as it is supposed to. If code was extremely easy to predict there would be far fewer hacks and errors. Unfortunately, understanding what exactly code will do in all scenarios is effectively impossible. An entire industry exists to audit code. Another industry exists to provide technical security in case things go wrong. Finally, yet another industry exists which assumes something will inevitably go wrong and require an enforceable resolution. This industry is composed of lawyers, courts, judges, and arbitrators. It occasionally uses police officers, detectives, and correctional officers as well.

Just because code isn't necessarily trustworthy doesn't make any particular blockchain code untrustworthy. As established earlier, trust is a function of economic incentives. Which parties have economic incentives relevant to the trustworthiness of autonomous blockchain code and DApps? Typically, the answer is some combination of the developers of the code, auditors of the code, potential hackers, base protocol maintainers (Ethereum miners in many cases), and possibly token holders since many of these applications have a native token used within the system. Hackers clearly aren't incentivized to help users, but their presence is important since they are the ones who exploit security vulnerabilities. For the others, there isn't a one-size-fits-all evaluation for how these

different entities impact the trustworthiness of various trust-minimized applications. They are almost certainly not equally trustworthy. However, there are some important things to consider which are generally applicable.

The incentives of the developers are very important. How directly do the developers benefit financially from successful usage of the protocol? Are the developers legally or financially accountable if something goes wrong? In today's blockchain apps, the answers are often unsatisfactory in both of these cases. Typically, rather than charge directly for services rendered, developers profit from releasing a token which is used in conjunction with the service. They own an outsized share of that token and benefit from its appreciation. However, token appreciation and successful usage have been shown to have only a loose relationship. Further, the developers are often not considered to be liable if things go wrong. There is no "throat to choke." Law enforcement can be called in to track down hackers, but it is widely accepted that nothing can actually be done to reassign stolen funds. There have been occasional breaches of this assumption, such as when Ethereum rewound the DAO hack, but this is considered against the ethos of crypto, or simply not possible in some cases.

The other parties have very limited, if any, ability to influence the trustworthiness of applications' on-chain code. When things go wrong, it is often widely reported and sometimes usage diminishes. However, this doesn't protect the users who were adversely affected by the first issue. Paid auditors with a reputation to protect is a good sign, although it is far from a guarantee that issues won't pop up. Base protocol operators have very little incentive or ability to ensure reliable service of a specific application using the base protocol. Token holders do have this incentive, but they are plagued by a tragedy of the commons and often have little ability to act if things go wrong.

At this point in time, Unbounded Capital would argue that traditional entities are much more trustworthy than on-chain code because of the economic incentives faced by the relevant businesses and individuals. This will likely only change if and when providers of on-chain code benefit more directly from the successful usage of their code and are more consistently held accountable for failures.