Bancor Is Flawed

ethereum (http://hackingdistributed.com/tag/ethereum/) bancor (http://hackingdistributed.com/tag/bancor/) ico (http://hackingdistributed.com/tag/ico/)

Monday June 19, 2017 at 10:18 AM

Emin Gün Sirer (http://hackingdistributed.com/egs/) and Phil Daian (http://hackingdistributed.com/pdaian/)

← Older (http://hackingdistributed.com/2017/06/15/town-crier/)

Newer → ()

Bancor just did their Initial Coin Offering (ICO) last week and raised a record $144M within a few hours. They now hold the record for the biggest crowd-funding, ever, in the history of mankind.

We don't want to dwell too much on what this illustrates about the current ICO craze. It's a fact that raising that much cash through a standard VC process would require a credible team, multiple rounds of funding, with much due diligence and milestones along the way. None of that happened here -- Bancor went from appearing on the scene 5 months ago to raising 9-digits cash with no demonstration that their scheme actually works.

In this post, we want to quickly make the case that their approach is flawed. To recap, they propose a scheme to provide liquidity for digital assets, using a smart contract. In essence, they propose a public algorithm by which they can always propose a bid/ask price for other people's coins. Now, a flawed approach doesn't mean that what they have is exploitable the way The DAO was, though we detail some immediate problems with the code below. What we mean is that the contract, as implemented, is far from meeting its purported narrative, even if no one takes advantage of the exploit.

The quick takeaway is that Bancor can be gamed by miners, and, even if the miners are naive or benevolent, will always trail the real market. It provides no efficiency guarantee during this discovery process, and will likely waste its reserves on market price discovery. You should think twice before you layer a coin on top of Bancor.

Let's do a quick walk through the red flags we encountered as we read the code and documentation for Bancor tokens, known as BNT.

Issues with Bancor Fundamentals

1. **Bancor uses lots of mumbo jumbo terminology.**

Bancor addresses the "double coincidence of wants" problem in economics through an "asynchronous price discovery" mechanism that blah blah blah buzzwords blah and more buzzwords. Half of these buzzwords aren't even real. "Asynchronous price discovery" is a franken-word they made up. It borrows asynchronous from distributed systems and "price discovery" from economics. The crowds eat this stuff up, but in reality, it's just a giant red flag -- there's no content here.

Let's move on, for it's possible to have a good idea underneath fluffy marketing.

2. **The core problem they want to address, "Double Coincidence of Wants" is not a real problem.**

"Double coincidence of wants" is a real problem in economics today in the sense that the "itsy bitsy spider" problem is a real problem in zoology -- that is, it's something one might learn in grade school, and it's completely irrelevant in the real world.

To be fair, you might indeed come to the market with two rabbits one day and I might come to the market with two chickens on the same day. I might also have an aversion to rabbit meat and a family history of mal de caribou. We would then be unable to conduct a trade.

In reality, this never happens, because...

3. **One can always use ether as a medium of exchange.**

There already exists a common currency through which we can trade. It's called ether, and we can use it no matter which token pairs we want to trade, because those very tokens are, by definition, implemented on top of Ethereum and were purchased with ether in the first place. Using BNT tokens is like stepping into a kid's swimming pool, placed in an ocean.

The entire point of the underlying currency, ether, is to serve as the medium of exchange for the diverse assets built on top. Sure, we can always layer BNT on top, and use BNT underneath the new Bancor-based tokens, but there would be little point in doing so besides creating a money flow for the people behind Bancor. For every coin they design that uses Bancor, we can design a more efficient version that doesn't use Bancor.

In short, the BNT tokens are for show. They are not necessary.

4. **Bancor is essentially a central bank strategy for automatically setting a dynamic peg for new coins.**

Behind the hype, Bancor comes down to a very simple idea: it's a smart contract that automatically provides a buy and sell price for a new coin. This is known as *a market maker*.

Let's suppose that you decide to create your own coin called X. You hype up the X ICO, sell $100M of X, and decide to back it with $10M of BNT. This is your reserve.

What Bancor proposes to do is to create a market for your coins by automatically buying and selling them for prices that would preserve the ratio of your reserve to the total supply. That is, depending on how much reserve the contract has outstanding, it will automatically offer a price for the X tokens. If the reserves are, say, $12M BNT, then it will offer to buy back X coins for higher prices, to bring the reserve fraction back in line to the usual level. If the reserves are low, it will offer lower prices. So you can always buy or sell into the contract, even if there is no one else to buy or sell to.

5. **That's it. That's the whole idea.**

It's only 40 lines of code.

Now, there is nothing wrong with raising $3.5M per line of code, if indeed there is a certain technical advantage that those lines of code possess.

6. **Everything Bancor can do for you on chain, you can do by yourself off chain.**

You yourself could easily have held that $10M in reserve, and you could have intervened in the market at your leisure. If you wanted to, you could follow the exact same algorithm as Bancor and achieve the exact same results. There is no value to be had from using Bancor.

Let us repeat that: in terms of controlling the price of the X token, there is absolutely nothing to be gained by using the fixed Bancor strategy. Whatever equilibrium price Bancor would achieve for you through its on chain management, you could have achieved while managing the reserve yourself, off chain.

To be fair, a Bancor proponent would say, "but you are committing provable reserves to your currency." To which the answer is, "there are many other ways of proving your reserves, none of which tie you down to a dirt simple and flawed strategy for life."

7. **It is much better to manage the reserves manually than to commit to the Bancor strategy.**

If you held the reserves yourself and issued buy and sell orders, like everyone else, you could not only follow the Bancor strategy to the letter if you liked, but you could use *any other* strategy that you preferred. You'd have the benefit of the immense neural network that you carry behind your eyes, and the freedom to switch strategies at any time.

8. **The Bancor strategy sets prices independent of the market**.

The prices that Bancor offers for tokens have nothing to do with the actual market equilibrium. Bancor **will always trail** the market, and in doing so, will **bleed its reserves**. A simple thought experiment suffices to illustrate the problem.

Suppose that market panic sets around X. Unfounded news about your system overtake social media. Let's suppose that people got convinced that your CEO has absconded to a remote island with no extradition treaty, that your CFO has been embezzling money, and your CTO was buying drugs from the darknet markets and shipping them to his work address to make a Scarface-like mound of white powder on his desk.

Worse, let's suppose that you know these allegations to be false. They were spread by a troll army wielded by a company with no products, whose business plan is to block everyone's coin stream.

What's your best strategy if you were in control of your $10M in reserves and were manually setting prices?

We don't know about you, but we'd hold the reserves tight and assuage the market. Maybe hold a press conference with the CEO, have the CFO show the books to an auditor, and present a sober and somber CTO describe how he was just making ginger bread houses with powdered confectionary for a fundraiser when he happened to sneeze. At the end of such a painful episode, we'd have battle scars, but also $10M in the bank.

If you used Bancor, your Bancor smart contract would have no knowledge of what is happening out there in the real world. It wouldn't know the market movements, it wouldn't know where the coin ought to be, and it would follow its blind strategy of offering bid/ask prices. That strategy involves making a market through thick and thin, without any information about reality. In fact, that reality is determined purely by external markets, and the contract will, unstrategically, use its reserves to discover and match what the markets demand of it at that instant.

In this case, Bancor would offer ever decreasing prices for X coins during a bank run, until it has no reserves left. You'd watch the market panic take hold and eat away your reserves. Recall that people are convinced that the true value of X is 0 in this scenario, and the Bancor formula is guaranteed to offer a price above that. So your entire reserve would be gone.

9. **The Bancor strategy fails to capitalize on excess value**

Now, a Bancor proponent would say "ok that was bad, but you're not giving us the full story. After your press release, people would start buying your coins, and the contract would rebuild its reserves."

Ok, let me illustrate what happens on the upside, and make the case that what they are saying is correct, but also incredibly inefficient. It's like saying "hey, your nerves do grow back after spinal injury." Yes, they do. Not so clear that you'll walk again.

Let's imagine that on the day of your big press conference, with the CEO, clean books, sober CTO and a new head of HR, your engineering team announces a new deterministic, asynchronous consensus protocol that takes a constant number of rounds [1], and a perpetual motion machine [2]. The crowds now love you. They go from dumping your shares at any price to fighting each other tooth and nail to buy your shares. In the time between ethereum blocks, a single X is worth infinity dollars on third-party exchanges. You could now literally sell just a single coin and retire forever.

But instead, you're using the Bancor strategy. The smart contract doesn't know that these changes are taking place. What it will do is offer to sell that first X for dirt cheap. It will fail to capitalize, on your behalf, on the difference between infinity and its sad, algorithmically determined low price. It just doesn't know. That extra money will get burned as miners' fees, as people try to outbid each other to own this token at the huge discount offered by the oblivious Bancor platform.

Essentially, we showed what happens when the true market price follows a step function between two extremes. The prices offered by the Bancor market making 40-liner will not follow this step function in tandem -- it is necessarily inefficient. That inefficiency is wasteful and gameable.

10. **The algorithmic dampening provided by Bancor isn't desirable for already liquid assets whose value is unstable.**

The previous two scenarios explored two extremes. But, to be fair, the following argument could be made in the steady-state case: "if your coins are locked up in this market maker, you lose the ability to insider-trade on the knowledge that most of the rumors about the CTO and CFO are false. However, this is simply a more general tradeoff between flexibility on one side and reliability on the other side -- although you deny yourself the flexibility to make extra profits on private knowledge, your users gain the comfort of knowing that there exists this market maker which will dampen price movements, and that it will not go anywhere."

This is true, the reserves do dampen price movements. If the price is going to be doing some ups and downs around a single mean value, Bancor can help facilitate trades by acting as a market maker.

But the situation is pretty bad when the price is leaving one level for another. If it's going down, then Bancor will bleed its reserves to keep the price close to the higher point that the price used to be at. And if it's going up, Bancor will sell coins at a price lower than the equilibrium point of the market, and therefore slow down the up movement.

11. **The Bancor strategy will not do anything to find or maintain the true equilibrium value of an asset.**

The preceding cases already illustrated the fundamental problem: Bancor is designed solely to maintain a reserve fraction. It has nothing to do with finding or maintaining the true value of an asset. It doesn't know, yet it'll blindly offer buy and sell bids. It'll use its reserves to discover what prices it ought to set.

12. **Bancor thus acts like a dynamically adjusted currency peg.**

Currency pegs have been tried again and again: ask any Argentinian for details. Any time you have a central bank trying to use its reserves to buoy up a peg, you have the opportunity for gaming. You'll recall that George Soros masterfully ran the reserves of the Bank of England down, simply by knowing their strategy. In this case, everything is happening on a public blockchain, using a fixed algorithm, with full visibility, while the true price is being set elsewhere through informed buy and sell orders.

13. **Bancor presents an arbitrage opportunity. It does not lead the market towards equilibrium, it trails the market, always and by definition.**

The mismatch between the Bancor price and the true market price is the cost that Bancor pays to have its smart contract be informed of the true value of the asset.

It is *not* the case that Bancor is helping the market perform price discovery. Quite the opposite: it's Bancor that is discovering the price, by virtue of offering buy and sell bids that are at odds with the value of the asset on the open market. It relies on arbitrageurs to bridge the gap and bring Bancor up to speed on what is happening. It pays a price out of its reserves for this function.

As such, Bancor will always trail the true value in the open market, and act as a buffer or a dampener. Bancor uses its reserve to be informed of the delta between the price it offers and the price out in the open.

14. **Bancor does not "eliminate labor" in price discovery.**

Despite Bancor's claims that they eliminate labor in price discovery, their current contract does nothing of the sort. It simply shifts the market maker labor onto arbitrageurs. It is now the arbitrageurs' job to notify the Bancor contract of the true price of an asset, and get paid a programmatic reward to do so.

15. **There is no indication that the Bancor strategy is an optimal, or even good, use of reserves to discover the price.**

The previous point was that Bancor uses its reserves to figure out where the market is, and sets a price accordingly. This isn't inherently bad, but it's neither what's in the advertised materials, nor is there any indication that the formula they used is the best use of reserves.

Why not, for instance, a market making strategy that approaches the target price using a different formula? Or uses AI techniques? Or uses past history of price action? The depth of the order book? The possibilities, for the strategy that a central bank would follow to manage its reserves, are endless. Bancor picked just the simplest possible option. The space of options remains unenunciated and uncharacterized, and there is no indication why this approach is superior to others in the Bancor materials.

16. **Bancor is a net negative in markets with substantial liquidity.**

Bancor-style automatic trading should be suspended when the external markets are already liquid. Throwing a dampener at a liquid market, one whose motions are preordained and predictable by all, is not the best use of those reserves.

The sensible way to design Bancor is to place some limits on how much of its reserves it will use in any time period, to avoid the problems we identified. There are currently no provisions for this. Doing this well requires importing some facts about liquidity from the external world into Bancor, perhaps with the aid of oracles such as virtual notary, town crier, oraclize and augur. But at a minimum, some limits on how much of its reserves the contract will spend in any given time period seem called for.

17. **Bancor claims to provide liquidity, but does not.**

Liquidity is the ability to buy or sell potentially large amounts without moving the price substantially. The Bancor contract does not guarantee this property, despite claiming that it does. Prices can move arbitrarily, and the price slippage is dependent on the amount bought or sold. This is a simple consequence of the fact that Bancor has no risk model and has no smarts. It's simply a market trailing dampener.

The preceding discussion examined the fundamental Bancor value proposition, in the abstract. Let's now examine its instantiation in Ethereum.

Front Running

18. **Bancor is open to front-running.**

Bancor's current implementation is open to a simple front-running attack. A miner, upon seeing that someone is submitting an order to buy from Bancor, would squeeze his own buy order ahead of the user's. He would thus always get a rate from the Bancor market maker that is better than what the user gets. Every time.

Bancor has a "minReturn" concept, akin to a limit order, that ensures that if the order goes below a certain level of profitability, the user can cancel it.

But the miners know exactly what limits the users set. Ethereum transactions aren't private. So a miner can squeeze in just the right order ahead of the user.

On the way down, front-running works the same way, with the miners squeezing sell orders ahead of the user, and thus pocketing the price difference.

And it's possible for miners to automate this process with a simple software kit. Further, even non-miners can take advantage of this behavior, by paying higher fees to appear before the Bancor transaction in the block.

19. **Bancor's suggested fix to front-running is broken.**

In a Twitter exchange, Bancor engineers mentioned that they are planning changes where they would charge the same fixed price to all transactions in the same block.

What they mentioned isn't currently implementable on Ethereum, at least not in a straight-forward fashion. Transactions within a block execute completely independently. If there are two transactions T1 and T2 in a single block, T1's execution occurs in isolation, without knowing about T2's presence in the same block. So it's not possible to charge T1 and T2 the same price, because T1 has no idea that T2 will execute in the future.

There are schemes we can imagine, where T1 and T2 are placed on the block first, and then a later "execute" transaction is placed on a subsequent block that looks backwards and gives the same price to T1 and T2. This gets around the limitation in the previous paragraph but it's really ugly and kludgy, not to mention more expensive to execute and open to new attacks of its own.

In general, any scheme that a) provides full information to miners b) doesn't include any nondeterminism and c) is vulnerable to ordering dependencies is gameable. Bancor and all their proposed modifications, including order floors and per-block prices, still satisfy all three.

Bad Math, Rounding and Lack of Testing

20. **Bancor reimplemented math.**

Bancor ended up reimplementing their own functions for arithmetic. That is, their own add, subtract, multiply, and exponentiation.

As an aside, this is sad to see for two reasons. First, finance applications should not have to worry about overflow errors. Ethereum should provide base types that make sure these kinds of reimplementations are never necessary in application code.

Second, no reimplementation of basic math routines should look the way Bancor's code looks. It's a mishmash of special numbers baked into the code. There is a certain style to writing code that is correct by construction. The code has to be crafted such that, not only is it correct, but it is easy to prove correct upon inspection. Bancor code is nothing like that, and baking magic numbers into the code is something for which we penalize first year undergraduates.

21. **Bancor did not test their own math.**

The sum total number of dedicated tests for these math functions is 6. Multiplication is tested solely by multiplying 2957 by 1740.

The coverage of the exponentiation function (https://pbs.twimg.com/media/DB9xe84XkAIAFGe.jpg:large) is abysmal. There are 0 directed tests that cover this critical function. There are other tests that take a quick path through exponentiation, but there are more than 30 different cases, and 30 different magic numbers embedded in the code. The existing, indirect, checks cover only a handful.

22. **Arithmetic errors can be fatal.**

Special magic constants litter Bancor code. So much so that we found it difficult to test this code for correctness ourselves. There is an art to writing correct, clean code, and Bancor exhibits none of it. An error in any one of the constants would be catastrophic.

And even simple rounding errors can be problematic in this game. A rounding error can enable an attacker to buy-then-sell a token to Bancor, and make a fraction of a cent in the process. If they can earn such a small quantity above transaction fees, then they'd do exactly that all day long to drain funds.

A final corner case to check for is what happens when the reserves are down to zero. The code needs to be able to recover from that scenario. The current code has special case handling for selling the entire supply. This seems strange for the continuous, path-independent Bancor formula.

Integration and Scale

23. **Bancor does not support the notion of supply caps for Bancor-based tokens.**

If the tokens that are based on Bancor are not actually securities, but serve as access tokens for a system, then they will typically correspond to some right to service. In many, though admittedly not all, scenarios, the system can only deliver a finite amount of service, so the designers will want to set a supply cap on their tokens. Yet Bancor's smart contract will create tokens out of the thin air in response to demand.

24. **Bancor does not scale.**

Bancor generates continuous on-chain tx volume for arbitrage on tokens that nobody cares about by definition. If they did care, those tokens would have liquidity and not need Bancor. It requires continuous on-chain activity for what is currently primarily off-chain economic action.

Users Overpay

25. **Bancor shortchanges its users.**

Since the Bancor contract cannot issue fractional tokens, it simply takes your money and gives you a number of tokens rounded (see rounding above) to an integer. Since you don't know when exactly your transaction will execute when you submit your bid, you have to pay both transaction fees to the miner to mine your transaction, and throw in an extra dollop of cash or coins to Bancor such that the contract can execute your trade without running under. If you guess wrong, you'll end up, say, getting 0.99998 coins, which conveniently rounds down to 0.

This will undoubtedly cause a lot of frustrated messages from users, who might feel that they submitted an honest bid, and got something short of their expectations.

Bancor whitepaper claims that you can predict what you will get back, but that's patently false: in the presence of concurent users submitting transactions, you cannot predict at all. Any extra amount you overpay is usurped by the Bancor contract.

Potential Reentrancy Issues

26. **Bancor code is "difficult to prove correct."**

That's code for "it's a mess."

Well-written code looks like a work of art. It doesn't matter if it's C or Go or Ruby or Prolog; in fact, it looks especially like a work of art if it's well-written C code. But it does matter if it's Javascript, because well-written Javascript code is like the mythical Yeti: often discussed, with snippets of evidence for its existence, but no one has seen it in its full, corporeal form.

The Bancor code has a distinct Javascript quality to it. This has been the hallmark of badly written smart contracts: they have messy code paths, don't follow best practices, and happen to work by the skin of their teeth. The code works on a good day with the wind on its back. But when it comes to corner cases, things get awfully complex. In the course I teach, when I point out such situations to students, they go into a diatribe that goes "well, you see, the problem can't arise because for it to happen, A has to happen first, but A is guarded by B, and C ensures that B cannot happen." This is a temporal logic proof, over a certain code path. It's complex, fragile in the face of code evolution, and totally unacceptable in professional code development. We want flat, simple invariants, maintained by following best practices. Without regard for the correctness of the student's code, we grade such cases a 2 out of 10. They ought to know better, and they're 20 year olds with two years of programming experience. Bancor definitely has to know better, given The DAO experience.

So, in what follows, we will point out violations of best practices as problems. We did not have time to look into whether they are exploitable, because it's tedious to do so and it's immaterial. There ought to be no violations of best practices in a good contract.

27. **Bancor code has a reentrancy problem in the sell() function**.

There is what appears to be a reentrancy problem (recall that an exploitable reentrancy bug affected The DAO) in BancorChanger. Now, not every reentrancy problem is an exploitable bug, and not every reentrancy bug gives rise to a The DAO like disaster. But nevertheless, there ought to be no state changes after a funds transfer, and there it is, right there on line 389 of the current (undeployed) code.

We cannot tell yet if this is exploitable, but it is disconcerting. There is no reason to perform a state change on something critical, such as virtualBalance, after the funds have been transferred.

28. **Bancor code has a different reentrancy problem in the change() function**.

There is a similar reentrancy problem in change(). When going from one token to another (recall the "double coincidence of wants!", perhaps it is referring to the double concidence of two different h4x0rs over the funds in Bancor), it performs, first, a buy() followed by a sell(). But buy() and sell() are both functions that call out of the Bancor contract. So it's possible to get into a messy situation if the token transfers during the buy() or sell() operation call back into the Bancor contract.

Again, we cannot tell yet if this is exploitable, but this is something that should be avoided. All transfers should happen after all the state changes.

29. **Bancor code assumes that ERC20 tokens based on Bancor are cooperative**.

The code assumes that the tokens layered on top of Bancor are cooperative. But ERC20 tokens can contain malicious code. It is difficult or impossible for Bancor engineers to vet the token code that is layered on top, and even if they do, it may be possible to change the ERC20 token contract after deployment.

Verdict

The Bancor code falls short of the narrative used to sell the code. Blindly making markets using a strategy that has no proof or reasoning for why it's good is a flawed idea. Additional problems, such as front-running, potential reentrancy issues, poor code quality, lack of testing and the general unnecessity of inventing a new currency, give us pause.

Keep in mind that there are lots of flawed ideas in the world. Not every flawed idea is catastrophic -- we've done many things that definitely were not proven good ideas, and survived to be writing these words. Perhaps miners will be altruistic and avoid front-running, perhaps all the magic numbers are correct even though they are untested, perhaps there are no rounding errors that matter.

But we know for sure that the Bancor approach is not the best use of one's currency. Someone armed with additional information not on the blockchain can certainly make markets in a better informed fashion. Since Bancor is possible to simulate off chain, committing to it on chain provides little to no additional value, except limit one's future moves.

Overall, it seems that the current Bancor approach is fundamentally inefficient and will bleed reserves. Assuming that there are no immediately exploitable issues in the code, and assuming that miners play nice and avoid front-running, we expect that we will see Bancor-based coins discover the limitations of the Bancor approach.

To end on a positive note, the optimistic way to look at the Bancor episode is that it's the first, flawed step in an interesting direction. It will surely be followed by more sophisticated approaches that have a better chance of living up to the hype. People are beginning to look for ways to codify Janet Yellen and place her on the blockchain. True, she need have no fear of losing her job right now. But sooner or later, we will have actualy smart contracts that provide strong guarantees as they manage a currency.

Footnotes

[1] This is impossible.

[2] This is also impossible.

Acknowledgments

We are grateful to Vitalik Buterin for his insightful feedback on earlier drafts of this document. Note that our technical analysis does not constitute investment advice. Crypto prices rarely follow rational rules, historically, some of the flawed systems that we have highlighted in this blog have done well, partly due to the increased attention they have received. Caveat emptor.

Recent Developments

- Bancor delays token activation (https://www.reddit.com/r/Bancor/comments/6i1uy5/token_activation_pushed_back_a_few_days/) and their switch to trading BNT on their own platform.

- The latest, hastily applied, patches (https://github.com/bancorprotocol/contracts/commit/5d651ea1d22b962a35d3acb375575530912dff20) to the Bancor code seem to revolve around rounding errors and precision loss.
- This post (https://keepingstock.net/hacking-the-bancor-protocol-for-fun-and-profit-6ddfb7f1cf56) discusses the rounding problem, mentioned above. It also mentions a social attack stemming from people not understanding how fractional reserves and the Bancor formula work.

← Older (http://hackingdistributed.com/2017/06/15/town-crier/)

Newer → ()

Share on Twitter (https://twitter.com/intent/tweet/?text=Bancor Is Flawed%20via%20@el33th4xor%0A&url=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/)

Share on Facebook (https://facebook.com/sharer/sharer.php?u=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/)

Share on Google+ (https://plus.google.com/share?url=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/)

Share on Linkedin (https://www.linkedin.com/shareArticle?mini=true&url=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/&title=Bancor Is Flawed&summary=Bancor just raised $144M through the biggest ICO in history. We describe why their approach is flawed.&source=http%3A%2F%2Fhackingdistributed.com)

Share on Reddit (https://reddit.com/submit/?url=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/&title=Bancor Is Flawed)

Share on Tumblr (https://www.tumblr.com/widgets/share/tool?posttype=link&title=Bancor Is Flawed&caption=Bancor Is Flawed&content=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/&canonicalUrl=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/&shareSource=tumblr_share_button)

Share on E-Mail (mailto:?subject=Bancor Is Flawed&body=http://hackingdistributed.com/2017/06/19/bancor-is-flawed/)

Emin Gün Sirer

Hacker and professor at Cornell, with interests that span distributed systems, OSes and networking. Current projects include HyperDex, OpenReplica and the Nexus OS. more... (http://hackingdistributed.com/egs/)

(http://hackingdistributed.com/egs/)

Follow @el33th4xor

Phil Daian

Phil Daian is a first year Ph.D. student at Cornell University, who is interested in cryptocurrencies and smart contracts. more... (http://hackingdistributed.com/pdaian/)

(http://hackingdistributed.com/pdaian/)

Subscribe

(http://hackingdistributed.com/hackingdistributed.atom) (https://www.facebook.com/egsirer)
(http://twitter.com/el33th4xor/)

Projects

- HyperDex (http://hyperdex.org)
- Weaver (http://weaver.systems)
- OpenReplica (http://openreplica.org)
- Nexus (http://www.cs.cornell.edu/people/egs/nexus/)
- Merlin (http://frenetic-lang.org/merlin/)

Recent Posts

Archive By Date (/calendar/)

- Bancor Is Flawed (http://hackingdistributed.com/2017/06/19/bancor-is-flawed/)
- Announcing The Town Crier Service (http://hackingdistributed.com/2017/06/15/town-crier/)
- Levels of Techie Enlightenment (http://hackingdistributed.com/2017/05/04/stages-of-enlightenment/)
- Hijacking Bitcoin: Routing Attacks on Cryptocurrencies (http://hackingdistributed.com/2017/05/01/bgp-attacks-on-btc/)
- Revealing the hidden links in the Monero blockchain (http://hackingdistributed.com/2017/04/19/monero-linkability/)
- A Thinking Person's Guide to the Latest Bitcoin Drama (http://hackingdistributed.com/2017/04/05/bitcoin-drama-response/)
- BitFury's Smallest-First Mining Is Bad For Bitcoin (http://hackingdistributed.com/2017/02/27/smallest-first-bad-for-bitcoin/)
- The Greening of Blockchains (http://hackingdistributed.com/2017/02/23/green-blockchains/)
- State of the Bitcoin Network (http://hackingdistributed.com/2017/02/15/state-of-the-bitcoin-network/)
- Miniature World: Measuring and Evaluating Blockchains (http://hackingdistributed.com/2017/02/10/miniature-world/)

more... (/page/2/)

Popular

- Introducing Weaver (/2014/12/16/introducing-weaver/)
- How to Disincentivize Large Bitcoin Mining Pools (/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools/)
- How A Mining Monopoly Can Attack Bitcoin (/2014/06/16/how-a-mining-monopoly-can-attack-bitcoin/)
- What Did Not Happen At Mt. Gox (/2014/03/01/what-did-not-happen-at-mtgox/)
- Bitcoin is Broken (/2013/11/04/bitcoin-is-broken/)
- Stack Ranking Is Not The Cause of Microsoft's Problems (/2013/08/24/stack-ranking-did-not-kill-microsoft/)
- How the Snowden Saga Will End (/2013/08/01/framework-for-surveillance/)
- What's Actually Wrong with Yahoo's Purchase of Summly (/2013/03/26/summly/)
- Broken By Design: MongoDB Fault Tolerance (/2013/01/29/mongo-ft/)
- Introducing Virtual Notary (/2013/06/20/virtual-notary-intro/)
- The Principled Documentation Manifesto (/2013/02/11/principled-documentation/)
- Introducing HyperDex Warp: ACID Transactions for NoSQL (/2013/02/05/hyperdex-warp/)

Blog Tags

bitcoin (http://hackingdistributed.com/tag/bitcoin/) / security (http://hackingdistributed.com/tag/security/) / hyperdex (http://hackingdistributed.com/tag/hyperdex/) / release (http://hackingdistributed.com/tag/release/) / ethereum (http://hackingdistributed.com/tag/ethereum/) / nosql (http://hackingdistributed.com/tag/nosql/) / selfish-mining (http://hackingdistributed.com/tag/selfish-mining/) / blocksize (http://hackingdistributed.com/tag/blocksize/) / dao (http://hackingdistributed.com/tag/dao/) / surveillance (http://hackingdistributed.com/tag/surveillance/) / privacy (http://hackingdistributed.com/tag/privacy/) / mongo (http://hackingdistributed.com/tag/mongo/) / broken (http://hackingdistributed.com/tag/broken/) / weaver (http://hackingdistributed.com/tag/weaver/) / nsa (http://hackingdistributed.com/tag/nsa/) / meta (http://hackingdistributed.com/tag/meta/) / leveldb (http://hackingdistributed.com/tag/leveldb/) / 51% (http://hackingdistributed.com/tag/51%25/) / smart contracts (http://hackingdistributed.com/tag/smart%20contracts/) / graph stores (http://hackingdistributed.com/tag/graph%20stores/) / bitcoin-ng (http://hackingdistributed.com/tag/bitcoin-ng/) / voting (http://hackingdistributed.com/tag/voting/) / vaults (http://hackingdistributed.com/tag/vaults/) / snowden (http://hackingdistributed.com/tag/snowden/) / satoshi (http://hackingdistributed.com/tag/satoshi/) / philosophy (http://hackingdistributed.com/tag/philosophy/) / mt. gox (http://hackingdistributed.com/tag/mt.%20gox/) / mining pools (http://hackingdistributed.com/tag/mining%20pools/) / micropayments (http://hackingdistributed.com/tag/micropayments/) / hyperleveldb (http://hackingdistributed.com/tag/hyperleveldb/)