namecoin

MONERO

Case Study: Alternate Blockchains

Jeremy Rand
Lead Application Engineer, The Namecoin Project
https://www.namecoin.org/

OpenPGP: 5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85

Presented at QCon London 2017

There are a lot of blockchains.

- Bitcoin is the most secure.
  - The oldest.
  - The highest commerce volume.
  - The most developers.
  - The highest market cap.
  - The most code review.
  - The most incentive to attack.
  - The most academic analysis.
  - The highest mining hashrate.

And yet people use other blockchains too.

- Why would other blockchains exist?
  - Most common modern reason: ethically dubious money-making schemes.
  - Blockchains are difficult to understand even for technically inclined people.
    - Weird chimeric combination of cryptography, distributed systems, economics, game theory, and some graph theory and politics mixed in.
  - Investors and end users almost invariably are incapable of evaluating blockchain technology details.
    - Yet investors and end users have somehow convinced themselves that blockchains will make them loads of money and/or make the Internet secure and/or overthrow the government.
  - Result: scammers create blockchains and easily acquire investors and users.

Are there legitimate non-Bitcoin blockchains?

- I think so.

In this talk:

- Use cases for non-Bitcoin blockchains.
- Case studies of two blockchains: Namecoin and Monero.
- Alternate approaches to Namecoin’s and Monero’s use cases.

Use case: naming systems

- DNS is centralized.
  - Drawbacks similar to centralized banking.
  - Domain names get seized by corrupt governments.
  - Domain names get hijacked/stolen by criminals.
  - DNS infrastructure gets DDoSed.
  - Court system to resolve disputes is expensive.

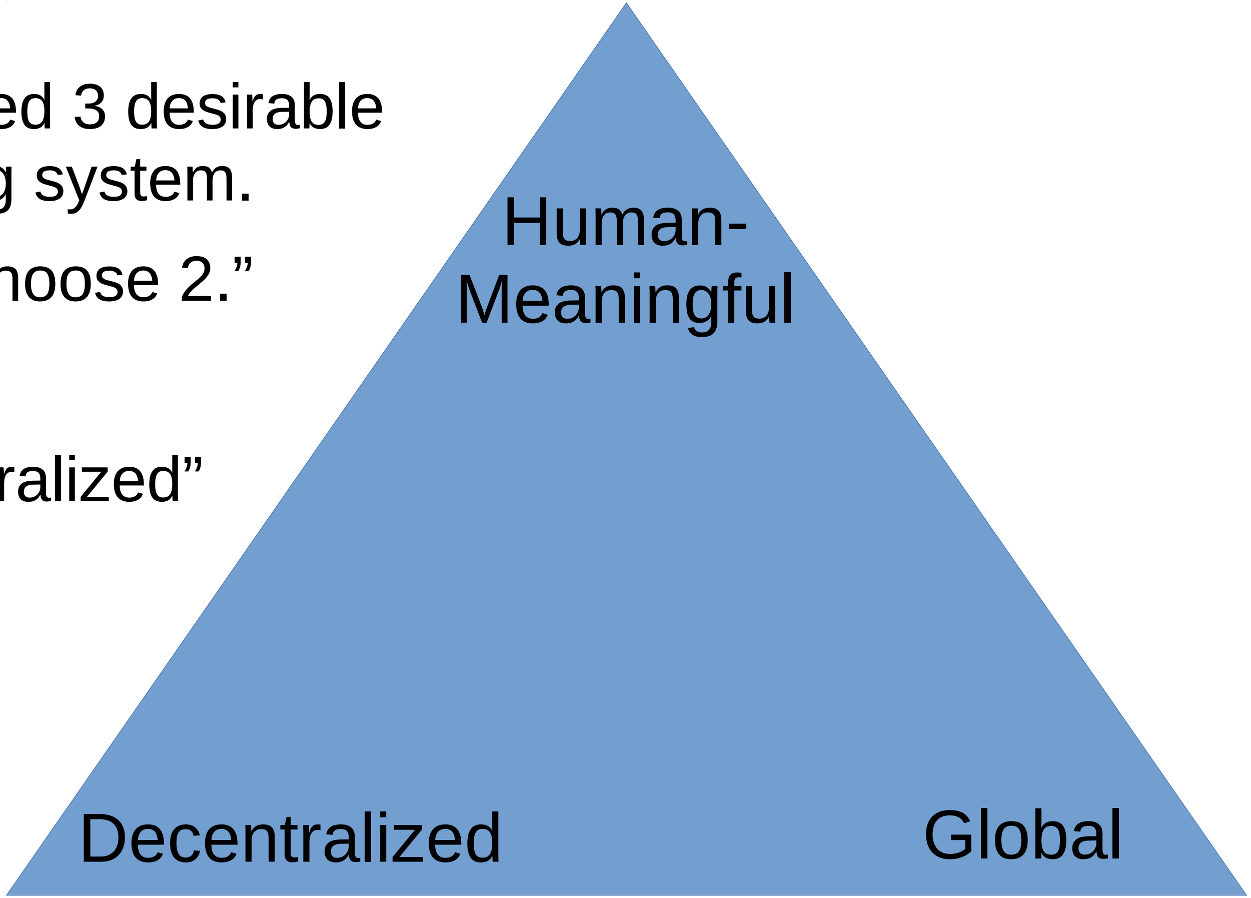
Use case: public key infrastructure

- TLS trusts thousands of certificate authorities.
  - CA’s get compromised.
  - CA’s achieve Too Big To Fail status.
    - Startcom, AKA the Martin Shkreli of computer security.
- You could use DNSSEC/DANE.
  - But then we’re back to the problems with DNS.

Zooko’s Triangle

- Zooko Wilcox formalized 3 desirable properties for a naming system.
- Zooko conjectured: “Choose 2.”

- DNS lacks the “decentralized” property.



Zooko’s Triangle == Decentralized Consensus?

- Decentralized global consensus was also believed to be impossible.
    - Lamport even wrote a math proof of its impossibility back in the 1970’s!
    - Yet Bitcoin solved it with the Nakamoto Blockchain.

- Dan Kaminsky and Aaron Swartz had a conversation that culminated in Aaron realizing that you could use a decentralized consensus system (a blockchain) to solve Zooko’s Triangle.
    - Aaron wrote up a proposal for such a system (Nakanames).

BitDNS

- Appamatto started a discussion in the Bitcoin community about “BitDNS and Generalizing Bitcoin”.
- One proposal was to add non-currency systems to Bitcoin’s blockchain.
  - The resulting blockchain, BitX, would include Bitcoin, BitDNS, and other systems.
- Bitcoin inventor Satoshi Nakamoto didn’t like this proposal.

Concerns with BitX

- Bitcoin is a social contract.
  - Users agree to all foot the bill of storing the Bitcoin blockchain.
  - Users get to use the currency that results from it.
- Filling the Bitcoin blockchain with non-currency data violates this social contract.
- “Piling every proof-of-work quorum system in the world into one dataset doesn't scale. Bitcoin and BitDNS can be used separately. Users shouldn't have to download all of both to use one or the other. BitDNS users may not want to download everything the next several unrelated networks decide to pile in either.” – Satoshi Nakamoto (2010 Dec 10).

Merge-Mined Sidechains

- Satoshi proposed a modified proof-of-work system that would allow miners to mine Bitcoin, BitDNS, and any other blockchain that might come later, without performance loss.
- This is called AuxPoW (auxiliary proof of work) or merge-mined sidechains.
- Satoshi argued that this would allow multiple blockchains to co-exist without being a danger to each other if many chains' miners ganged up on one chain.

Namecoin

- Namecoin is the implementation of the BitDNS proposal.
  - Released by Vincent Durham (a pseudonymous author who later disappeared – much like Satoshi).
  - Includes AuxPoW (though it didn't have AuxPoW at launch).

Structure of Namecoin

- Fork of Bitcoin Core with minimal changes.
- Names are just coins that have a special scriptPubkey.
- Adds 3 new script opcodes: OP_NAME_NEW, OP_NAME_FIRSTUPDATE, and OP_NAME_UPDATE.
- Name outputs push a name opcode and some arguments (e.g. name and value) to the stack, and then immediately drop them again before the rest of the scriptPubkey.
- That means that name transactions are actually valid Bitcoin transactions (the push/drop stuff at the beginning is a NOP in Bitcoin).

Name registration workflow

- OP_NAME_NEW creates a salted commitment to the name being registered.
- OP_NAME_FIRSTUPDATE reveals the name being registered and the commitment salt.
  - Only valid if spending an OP_NAME_NEW input that has a matching commitment.
  - Only valid if the OP_NAME_NEW input is at least 12 blocks old.
- This workflow makes sure that attackers can't front-run registrations as soon as they see them appear on the network.

Name data

- Names consist of a “name” argument (OP_PUSHDATA of up to 255 bytes) and a “value” argument (OP_PUSHDATA of up to 520 bytes).
- OP_NAME_UPDATE can do either of the following:
  - Change the value of the name.
  - Transfer the name to a new owner.
- Names expire 36000 blocks after their most recent OP_NAME_UPDATE.
  - To renew a name, you can use OP_NAME_UPDATE and use the same value it already has.
- Global uniqueness of **unexpired** names is enforced as a consensus rule.

Name consensus rules

- Global uniqueness of **unexpired** names is enforced as a consensus rule.
- Immutability of the **name** field is enforced.
  - This actually was broken on initial release.
  - Michael Gronager from Kraken reported this to us, and demoed using this to forge name updates.
    - The issue was fixed quickly; no real-world attacks occurred.
- The **value** field has no consensus-critical rules at all.
  - Interpreting values as DNS records is entirely by convention.

Namespaces

- Namecoin is a data-agnostic key/value store.
- By convention, names begin with a namespace.
  - Namespaces end with a forward-slash.
- The domain name “example.bit” would have the Namecoin name “d/example”.
- The identity “jeremy” would have the Namecoin name “id/jeremy”.
- Nothing about namespaces or their semantics is part of the consensus rules.
  - This adds flexibility for implementing new use cases.

Namespaces and Satoshi's BitX Opposition

- Namecoin can be used for different use cases, e.g. domain names and identities.
  - Does this make it similar to BitX?
- Namecoin's position is that creating a new chain makes sense if and only if you want different validation rules than existing chains.
  - So Namecoin can be a single chain, but BitX can't.
- There's plenty of room for debate on this.

Does AuxPoW deliver security?

- Security issue: the parent chain (Bitcoin) and the child chain (Namecoin) can be attacked independently of each other.
- Bribing unethical, economically rational miners to 51%-attack Namecoin is much cheaper than doing so to Bitcoin.
- Bitcoin mining yields 12.5 BTC * 1060 GBP/BTC = ~13,250 GBP every 10 minutes.
- Namecoin mining yields 25 NMC * 0.000307 BTC/NMC * 1060 GBP/BTC = ~8 GBP every 10 minutes.
  - That's 292,878 GBP to steal a name by censoring its renewals until it expires.
- (Market data from Bitsquare.)

Real miners probably do have ethics

- So maybe it would be more difficult to 51% attack Namecoin than the previous slide estimated.
- But relying on ethics isn't what we want.

Namecoin 51% Incidents

- 3 different mining pools have possessed a majority of Namecoin hashrate in the past. Each had circa 80% for some time.
- BTC-Guild only mined empty blocks – caused transactions to take circa an hour to be mined.
- GHash.IO never attacked Namecoin – but a rogue employee there did do a double-spend attack on Bitcoin.
- F2Pool actively funded Namecoin development – more on F2Pool later.

Closely following upstream is critical

- In July 2015, Bitcoin activated the BIP66 softfork.
- Pieter Wuille then disclosed that BIP66 fixed a consensus bug in OpenSSL that could cause a chainfork.
- Just one problem: many other chains, including Namecoin, hadn't yet activated BIP66.
  - Zeroday drop!

Panic Mode: On

- We already had about 60% to 80% of hashrate (i.e. only F2Pool) supporting BIP66; 95% was needed to activate.
- Most of the miners were very responsive when we notified them, we reached around 92% quickly.
- And then we couldn't reach the remaining miners.

The Nuclear Option

- As soon as we saw Pieter's disclosure, we briefly discussed the idea of asking F2Pool to activate BIP66 enforcement early if we couldn't reach the other miners.
- After we had no luck getting up to 95%, Wang Chun from F2Pool independently had the same idea and offered to do an early softfork.
- This would risk network instability short term, but we had already instructed users a week prior to not register names or make trades until things were resolved.
- So a few short chain reorgs were probably not going to hurt – and if we didn't do that, anyone could cause a chainfork at any time that wasn't going to resolve itself.

How did the Nuclear Option work out?

- F2Pool announced the early softfork 2 days in advance.
- BitMinter started enforcing the softfork shortly after F2Pool did.
- A short chainfork occurred; an 11-block-long chain tip was orphaned when the early-softfork chain overtook it after a bad-luck streak by F2Pool ended.
- We hit the 95% threshold very quickly after the early softfork began, meaning that the softfork activated for all nodes – we were safe.

So did we do the right thing?

- We were really, really uncomfortable with this at the time.
- It's not good to set a precedent that it's acceptable for a few developers and 1 large mining pool to activate an unplanned softfork.
- However… after we did this, an altcoin that hadn't aggressively activated BIP66 (Peercoin) was attacked with Pieter's zeroday, and their chain forked for more than a week.
- In retrospect, I think we picked the least bad option available. A 2-hour-long chainfork with 2 days of advance notice is way better than an unplanned week-long chainfork.

But let's not have that happen again.

- The advice from Luke Dashjr after that whole mess was: if Bitcoin makes a change, follow their lead even if it looks non-urgent, and follow their lead quickly.
- If we had activated BIP66 around the same time as Bitcoin, we wouldn't have been put in a bad situation.
- Your upstream will not give you advance warnings of security fixes that they're deploying or disclosing.
- If you're not prepared to quickly merge upstream commits, your blockchain is going to have bad things happen.

Why hadn't Namecoin activated BIP66?

- Feature creep.
- Early in Namecoin development, Namecoin devs started adding unnecessary patches against Bitcoin.
  - A custom Windows build system.
  - A rewrite of the RPC subsystem.
- Namecoin devs also didn't merge lots of Bitcoin commits.
- Result: it was impossible to keep up with Bitcoin innovations.
- (The current Namecoin developer team wasn't around when this started.)

Solution? Rewrite the code from scratch!

- F2Pool bailed us out by contracting our Chief Scientist (Daniel Kraft) to do a from-scratch rebase against upstream Bitcoin Core.
  - Namecoin Core has a very different philosophy about patches against upstream.
  - The **only** cases where we merge code that isn't in Bitcoin Core are when the code wouldn't be applicable to Bitcoin because of the different use cases (currency vs naming system).
- At the time BIP66 activated for Bitcoin, Namecoin Core had passed QA testing by F2Pool but the other pools hadn't adopted it yet.

Lesson learned

- Don't deviate from your upstream blockchain codebase unless you have a very good reason to do so.
  - If you want a feature that isn't relevant to your blockchain's intended use case, get it merged upstream first.
- Even if you diligently attempt to follow this philosophy, you will encounter cases where you need to maintain patches against upstream code.
  - Try to eliminate these patches as soon as possible.

AuxPoW Specification Quirks

- Merge-mined sidechains include a parent block header and a parent block coinbase transaction in the sidechain's block headers.
  - Problem: coinbase transactions often include hundreds of outputs to pay mining pool users.
  - Even SPV validators of the merge-mined sidechain need to process this.

P2Pool's AuxPoW

- P2Pool (a decentralized Bitcoin mining pool implemented as a merge-mined sidechain) uses a hash midstate proof to get rid of most of this data.
  - Adopting this in Namecoin would be a hardfork.
  - Not much peer review.
  - Not much battle-hardening – breaking P2Pool doesn't yield much payout.
  - Cryptographic implications of allowing attackers to choose hash midstates are difficult to evaluate.
  - We're taking the “wait and see” approach.

Luke Dashjr's AuxPoW

- Luke's work on pegged sidechains includes a new AuxPoW spec.
- Solves the coinbase transaction bloat issue.
- Doesn't add crypto assumptions like hash midstate proofs.
- Requires the parent chain (i.e. Bitcoin) to be merge-mined.
  - That's a Bitcoin hardfork.
  - We're not optimistic that this will happen anytime soon.

AuxPoW Specification Quirks (2)

- Merge-mined sidechains' block headers abuse the 32-bit block version field.
  - Bit 0x100 is used to signal that a block is merge-mined.
  - The top 16 bits are the “chain ID” (used to allow mining multiple merge-mined sidechains at once).
- Problem: Bitcoin's BIP9 (VersionBits) uses these bits for different purposes.
  - Namecoin plans to hardfork to fix this.

Layered Chains

- In response to some of these drawbacks of merge-mined sidechains (specifically Bitcoin), Blockstack (a Namecoin competitor) tried a different approach: layered chains.

Layered Chains: split consensus layers

- You can use OP_RETURN outputs in a parent chain to create transactions (additional data can be stored externally, e.g. in a DHT).
- This uses the parent chain to provide secure ordering of transactions.
- A second consensus layer decides which of those transactions are valid.
  - 2nd layer is invisible to the parent chain.
- This is implemented in Blockstack, using Bitcoin as its parent chain.

Are layered chains secure?

- Against reorganizations, they have much better security than merge-mined sidechains.
  - Re-orging Blockstack means you have to re-org Bitcoin, and vice versa.

Layered chains and lightweight clients

- Problem: for the 2nd layer to work properly, you need to download the full parent chain (the parent chain doesn't enforce the 2nd layer's rules).
  - Blockstack nodes need to download the entire Bitcoin blockchain in order to apply Blockstack's consensus rules.
- Headers-only SPV nodes are not possible with layered chains.
- Blockstack's “solution”: rely on trusted checkpoints.
  - Not secure or decentralized at all.
- As of circa a month ago, Blockstack's CTO told me that they've started telling users to download the full Bitcoin blockchain.
  - Their website still talks about the trusted checkpoints … mixed messaging?

Are layered chains realistic?

- Making SPV impossible is likely to drive users to less safe lightweight clients, i.e. trusted 3rd party systems.
- Blockstack's developers operate trusted 3rd party systems.
- Cynics may see this as an attempt to look security-conscious while getting lots of customers for their insecure paid services.

Do layered chains scale?

- Blockstack rate-limited their transaction volume to 20-30% of Bitcoin volume.
  - Using 20-30% of Bitcoin blocks violates Bitcoin's social contract.
  - Bitcoin users lose 20-30% of their intended capacity (paying higher fees as a result).
  - Bitcoin users have to carry around Blockstack data forever.

What do the Blockstack devs recommend to make their system scale?

- “the community needs to look into side chains” – A paper by the Blockstack founders.
  - Remember, Blockstack was founded because they claimed merge-mined sidechains were insecure.
  - Makes sense....

Blockchain use case: Anonymity

- Bitcoin has terrible privacy.
- Transaction graph analysis can deanonymize Bitcoin payments.

Monero: an anonymous currency blockchain

- Monero is a completely new codebase – no common code to Bitcoin.
- Monero signatures are ring signatures (hides the sender), not standard ECDSA.
- Monero addresses use ECDH (hides the receiver), not standard public key hashes.
- Many things are redesigned compared to Bitcoin.

Why would you want to redesign Bitcoin?

- Hedge in case Bitcoin's crypto (e.g. curve choice) gets broken.
- Allows experimental changes to be tested in the field.
- Monero markets itself as a science experiment, not a store of value.
- Hardforks scheduled every 6 months – advances to the next phase of the science experiment.
  - Would be disastrous for Bitcoin, but beneficial for a small experimental coin like Monero.
  - Monero's experiments can inform future Bitcoin changes.

Could Monero's features be added to Bitcoin?

- Probably not.
- Monero's ring signatures don't scale as well as Bitcoin's ECDSA.
  - Hiding the spent status of coins means you can't prune spent coins.
- This makes it an excellent use case for an alternate blockchain.

Other approach: contract chains

- Contract-oriented chains like Ethereum could be used instead of a dedicated chain like Namecoin or Monero.
- Potential issue: the social contract of Ethereum isn't tied to a single contract.
- If your contract has a security bug, it's considered equivalent to MtGox rather than the blockchain.
  - That means you probably won't get a consensus fork bailout.
  - And if you do, some users may boycott the fork, like Ethereum Classic.

Other approach: pegged sidechains

- Proposed by Blockstream.
- Chains that can temporarily hold coins from other chains.
- It's unclear how the pegging mechanism would work in a decentralized way without sacrificing security.

Exchange rate volatility

- Chains like Namecoin and Monero suffer from increased exchange rate volatility (since they have their own currency).
- If your exchange rate is too low, you become cheap to attack.
- Approaches like layered chains, contract chains, and pegged sidechains avoid this by using an existing currency.
- However, pegged sidechains don't have a block subsidy to incentivise security (besides transaction fees).
  - Bitcoin won't either in the future, but it will probably have a very high exchange rate by then.

Some final thoughts

- Forking the Bitcoin code and safely maintaining that fork takes a lot of specialized expertise.
- It is rare to find use cases where an existing chain (usually Bitcoin) can't do what you want.
- Most attempts to create a blockchain don't go well.
- If you attempt it, make sure your users are aware of the risks of using experimental code.
  - Yes, your new blockchain's code is experimental.

Some final thoughts (2)

- The blockchain field has scams everywhere.
  - Some scams are unintentional.
  - It's important to call out scams and broken designs where you see them.
- The Monero and Namecoin developers regularly get complaints: “Why won't you just collaborate with other projects?”
  - We love collaboration. (E.g. Monero is working on a naming system, and is collaborating with Namecoin on it.)
  - But we also want end users to be aware of what they're getting. Chains with slick marketing but no innovation are ethically problematic.

Some final thoughts (3)

- If your idea for a new blockchain can be marketed as “this is an interesting experiment that might yield a really cool system in a few years”, there's a decent chance you're okay.
- If your idea for a new blockchain is marketed to investors as “this will revolutionize <Insert Industry Here> and is therefore a good investment”, there's a strong chance that you've overestimated the technical soundness of your hypothetical blockchain.

Some final thoughts (4)

- No one benefits from the blockchain field being famous for ill-conceived products that lose people's money.
- Blockchains are hard; people **will** lose their money sometimes.
- Transparency about security tradeoffs goes a long way to ensuring the credibility of our field.

Thanks for having me here!

- https://www.namecoin.org/
- OpenPGP:
  5174 0B7C 732D 572A 3140 4010 6605 55E1 F8F7 BF85