A Paradigm Shift in Blockchain

Saito is a fundamentally new class of blockchain. The differences start with the problems Saito solves. Other blockchains focus on...

1. How to Make Block Production Difficult.

Proof-of-work (POW) and proof-of-stake (POS) blockchains make block production difficult and give all of their funds to the block producer. Attackers who can buy or rent 50 percent of the "work" needed to produce blocks can collect up to 100 percent of network revenue, or even more if they are willing to launch double-spending attacks.

In consensus systems like Tendermint where validators determine which blocks are valid, the cost of attack is technically even lower and networks can be forced into paralysis by attackers with only 34 percent of network stake. Similar problems with majority dominance occurs in DAG structures. Economic forces incentivize and encourage collusion in these networks.

Making matters worse, the techniques these networks use to make attacks "difficult" rely on markets (for hashpower or capital) which exist outside the control of the blockchain. Not only does this make most networks susceptible to basic economic attacks (see "discouragement attacks" etc.), but the existence of the blockchain itself incentivizes the commoditization of the supply curve for the block-production resource. Economic forces unleashed by the blockchain undermine its security over the long-run.

SAITO GUARANTEES THAT ATTACKS ARE ALWAYS EXPENSIVE

Saito recognizes that all forms of "difficulty" are reducible to economics. This is why it solves a different problem: guaranteeing that it is always expensive to produce blocks. The network accomplishes this by separating the block reward from the block producer and using a provably-fair lottery to reward nodes in proportion to the amount of money they collect for the network. Honest nodes get paid by processing user transactions. Attackers have to spend their own money.

Because the way the lottery works, producing the longest chain of blocks requires attackers to spend more money than they can ever recover in fees. And because expected losses increase as transaction volume grows, the network can defend itself very easily: spending the money contributed by attackers on increased transaction flow that speeds up their transfers of wealth to the network. Attacks are truely difficulty because attackers must inevitably go bankrupt.

Guaranteeing that attacks are always expensive eliminates 51 percent attacks and changes the security dynamic of the blockchain completely. Security is more than double that of existing POW and POS networks. There are no more incentives for participants to collude. All nodes in the network can be paid for their help with fee collection, and the network remains open in all situations. Users can meanwhile continue to get quantifiable security by waiting the appropriate number of confirmations until re-writing the blockchain costs more money to attackers than any potential double-spends are worth.

2. How to Hide the Volunteers in their Systems

All non-Saito blockchains have volunteers hidden in their systems. They have volunteers hidden in the form of the businesses which collect transactions from users but somehow never think to sell that transaction flow to block producers. Volunteers in their peer-to-peer network must prevent block producers from hoarding or censoring transactions. POS implementations that pay block producers need volunteer validators to keep them in-line. POS implementations that pay validators need volunteers to propose blocks.

Data storage is also thrust unapologetically into the hands of volunteers. BSV and BCH actively discourage miners from storing blockchain data. DAGS like IOTA introduce "masternodes" which are supposed to store data, but are unpaid for the service. Hashgraph advertises itself as a data network, but encourages nodes to reduce old transactions to hashes of the transaction data. Where can a user go to actually get the data other users are pushing into the blockchain? The current craze in application networks like Blockstack is meanwhile push data off-chain onto... volunteer networks like IPFS. What is the thinking: if our volunteers won't do it, maybe theirs will?

As these networks scale, the costs to volunteers necessarily rises, pushing these networks towards technical non-solutions (lite-clients!) that decrease the cost burden on some users while shifting costs to volunteers elsewhere in the system. But they cannot do otherwise: miners are stakers are strictly incentivized to fund their revenue-earning function. At best these networks can create insecure subsidy mechanisms controlled by developers, or complicated systems with multiple payments to multiple parties and entire classes of attacks on their artificial pricing mechanism.

SAITO GUARANTEES THAT ALL NODES ARE PAID FOR THEIR VALUE

Saito introduces and has patented the technique of adding cryptographic signatures to blockchains on their network layer. This allows the consensus mechanism to measure the actual value that routing nodes provide to the network. This is used to pay them in proportion to the value they contribute to the network: there is no need for volunteers in the businesses that connect to users or those that form part of the routing network.

Automatic transaction rebroadcasting (patent pending) eliminates the needs for volunteers to provide data storage, since nodes that do not store blockchain data are incapable of producing new blocks. Users who wish to rely on unreliable volunteer-provided storage systems or layer-two networks are of course still welcome to do so. Users who need reliable on-chain data service are able to pay for it at prices that reflect the actual market cost of storing their data.

This last point is a major point of departure from conventional blockchain economics, which have developers carve up revenues themselves: the Saito consensus mechanism unleashes market pressures within the block-creation process which auto-adjust the payments made to routing nodes and block producers so as to maximize the revenue collected by the network. There is no need for insecure and arbitrary developer-imposed subsidy mechanisms which lower security, or complicated schemes that require "validators" or "proposers" or "masternodes" to perform critical network services or whitepapers which mask the fact that these services are unpaid. The network pays for everything it needs and scales payments to bottlenecks as needed: growth pays for itself.

3. How to Use Smart Contracts to Create Web 3.0 Platforms

The vast majority of "application blockchains" under development are smart-contract platforms. These systems introduce coordination problems (through sharding), network bloat (through their inability to incentivize data storage), monopolization (through their lack of funding for API access points) and add sharp limits to the scalability of the underlying blockchain as computers in the network need to spend CPU cycles to maintain program state.

SAITO SIMPLIFIED APPLICATIONS AND ELIMINATES BLOAT

Saito is a pure-play PKI network. Users send transactions into the network, which broadcasts their data to all recipients. Developers build applications on the edge of the network, much like in the traditional Internet. Applications can maintain consensus simply by tracking the data broadcast on the data-layer, making it possible to program smart contract systems and meta-blockchains. Upgrading these networks is as simply as upgrading a software module.

Saito's approach eliminates bloat on the network layer. The protocol is as simple as possible: a UTXO set that can be spent to pass signed byte-arrays to peers. What the computers on the edge of the network do with this data is not a concern of the blockchain, which concerns itself with economic efficiency and self-sustainable economics. Scaling problems associated with smart contracts fall away as there is no need for all of the computers in the network to process every transaction. Sharding is possible and simpler to implement.

Beyond simplifying how applications are developed and deployed, Saito's consensus mechanism further lowers their cost by improving the efficiency of the network in ways that smart contract systems cannot provide. Saito makes it unprofitable to run sybil nodes (or permit oneself to be sybilled). The network also rewards participants who create efficient routing paths between users and block producers. The costs of running the network fall and savings are passed along to the users and applications that pay the transactions fees for use of the network.