Using the Chain for what Chains are Good For

This proposal is for a high-level introductory talk. The list below can be compressed into a paragraph if that fits the abstract format better.

The blockchain is simultaneously Bitcoin's core innovation, letting it succeed where no other system had before, and its greatest weakness, requiring miner discretion in choosing transactions, while no other part of the system has any third-party dependence. In exchange for this dependence, miners produce an increasingly-immutable proof-of-publication medium, allowing anyone at any time to see what transactions occurred and in what order, and to be assured that their view matches the view of all other validators.

Bitcoin's essential use of the blockchain is to prevent *double-spending*: to publish transaction outputs as they are created and later consumed as inputs. This provides an unambiguous beginning and end of each output, which an otherwise unattainable goal in a relativistic world. However Bitcoin uses the blockchain for much more than this: it has a script system which allows users to set arbitrary spend conditions on their coins; it allows transactions to be time-locked and invalid until some time has passed; it ensures transactions are executed atomically and not peeled apart on the network. All of these conditions are published on the chain and verified by all validators. It is the thesis of this talk that these "non-essential" uses of the blockchain can often be done with significantly reduced (or eliminated) use of the blockchain, and that this has tremendous benefits for the transactors themselves as well as the network as a whole.

First, we describe the costs of blockchain usage.

- Blocks appear only every ten minutes on average, meaning long and unpredictable latency for users of the chain.
- During this time transactions are published to the network, leaking private timing and source information, plus the transaction data itself, to anyone who cares to analyze it. This exposure undermines users' privacy, businesses' confidentiality, and the fungibility of the currency itself.
- This public data can be seen by miners before they include transactions, which poses a censorship risk for users as well as an incentive for adversaries to pressure miners into censorship.
- Blockchain space is limited, forcing users to pay for this data even as its existence harms them.
- This data must be validated by all participants in the system who want to verify that their view of its state is untampered with. Since supporting these users is a core component of the Bitcoin ethos, the result is a limitation on the scalability of the entire system.
- Finally, the rules for validating data on the blockchain are system-wide rules that cannot be changed without agreement from all users. This is not even possible for conflicting rules, but when it is possible it is extremely hard to measure and hard to achieve.

Next, we look at ways to avoid using the blockchain so heavily, which are largely the subject of new and ongoing research. These exciting developments evade all of the listed problems by simply avoiding the blockchain except where global agreement on double-spends is needed.

One idea in this direction is *payment channels*, in which coins are first placed into multisignature outputs jointly owned by two parties. Those parties then send money between each other by updating (but not publishing) a transaction which spends the coins in the channel. By structuring every update as a valid blockchain transaction, each party is able to close the channel by simply publishing the current state, while the multisignature ensures that neither party can double-spend without the other's cooperation. An extension of payment channels is the *Lightning network*, which links together multiple channels, allowing payments to be routed through many hops by atomically executing channel updates in many channels.

Payment channels work when there are many transactions between the same set of participants, who don't want to appeal to the blockchain every time. However, the transactions themselves contain extra data describing the participants' public keys, conditions for what happens if one party publishes an expired state and data to atomically link the different channels together. Separate from (but complementary to) payment channels are several techniques for removing data from the transactions themselves.

An pair of old ideas are *sign-to-contract* and *pay-to-contract* in which data is committed to by signatures or public keys that would be included in the blockchain anyway. These commitments take no extra space and cannot even be seen except by validators with access to auxiliary data. The data is attached to specific outputs so in the case of colored coins, for example, it takes advantage of Bitcoin's native double-spend protection.

More advanced is to use *key aggregation*, a technique which requires a new signature type in Bitcoin, to do multisignatures. With key aggregation, the different parties in a multisignature transaction are able to create a single combined pubkey which requires all of their participation to sign for. The resulting key and signature are indistinguishable from the single-signer case, providing extra privacy, saving blockchain fees, and lessening the load on validators.

Building on key aggregation, an algebraically related idea is to replace hash-preimage challenges with *adaptor signatures*. These are signatures which, when combined with some auxiliary data, reveals the discrete logarithm of some elliptic curve point. These allow transactions or payment channels to be linked atomically, but again without any extra blockchain space. It is impossible for non-participants to tell that the transactions are linked, or even that they are linked to anything at all!

Finally we touch on *zero-knowledge contingent payments*, first invented by Greg Maxwell and improved by several others. This is a way to put arbitrary conditions (except that they must be satisfied by a person chosen in advance) on the spend of a coin, again in such a way that the conditions are invisible to validators of the chain.