Open Design Question: Transaction Signing  
  
**Description**

Satoshi taught us that a blockchain can solve many protocol problems: software can easily check arbitrary block validation rules, and a simple selection rule (cumulative difficulty) can force a unique protocol history (onto participants with varying degrees of network participation).

Yet blockchains only carry information one way. We can use our phones to alter the blockchain (with a Bitcoin transaction), but the blockchain all by itself can’t alter our phones. To do this we must run applications on our phones that are designed to use blockchain data, for example a ‘Block Explorer App’ that displays transaction data, or a ‘Bitcoin Call App’ that makes a pre-specified phone call when a certain transaction is received. Similarly, the Truthcoin blockchain can read the Bitcoin blockchain (making it easy to accept Bitcoin deposits), but alone cannot write to the Bitcoin blockchain (making it more difficult to authorize Bitcoin withdrawals). Completing the implementation will require a tool analogous to ‘Bitcoin Call App’ which (probably) watches Truthcoin for withdrawal requests and signs/broadcasts them.  
  
Thus the problem: Precisely how should a distributed blockchain manage this application, and successfully sign Bitcoin transactions?

**Working Solution**

While I don’t know what solution will eventually be used, I can work with one of the several working solutions I proposed in my original whitepaper.

Oracle watches the block, checks for enforced rules. One rule builds the unsigned transactions and embeds within block.

Withdrawal transactions have a Bitcoin fee.

Zzzz

Once a block has 20 (or so) blocks on top of it, the oracle jumps in and signs the transaction. This way we don’t need to worry about orphans. A 20 chain reorg would be pretty screwed up anyway.

ZZZZZZ