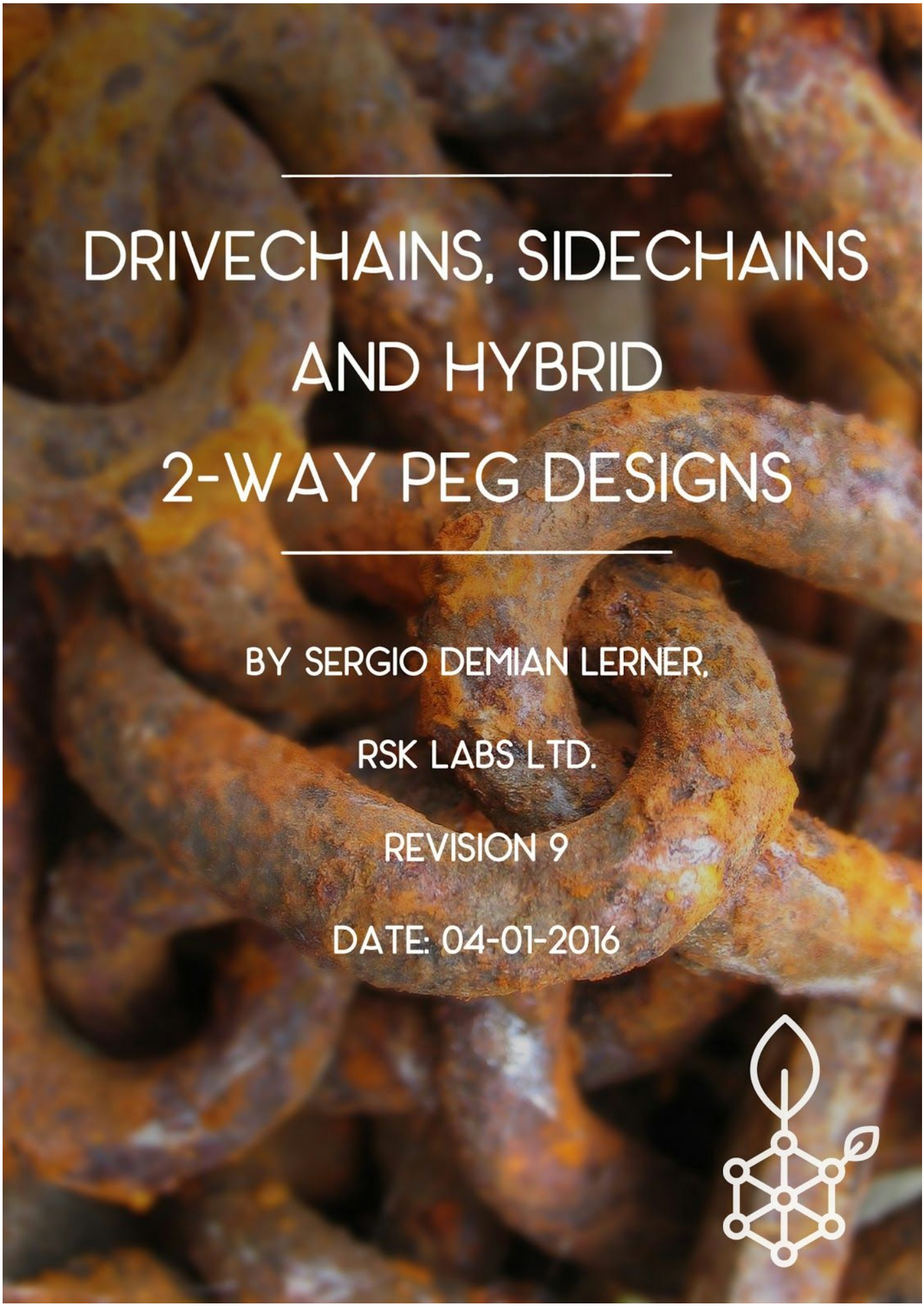


DRIVECHAINS, SIDECHAINS AND HYBRID 2-WAY PEG DESIGNS

BY SERGIO DEMIAN LERNER,

RSK LABS LTD.

REVISION 9

DATE: 04-01-2016

Index

- Index
- Introduction
- The 9 Kinds of Magic
  - Who Has Custody
    - External party or group has the custody
    - Bitcoin miners have the custody
    - Bitcoin stakeholders have the custody
  - What kind of Control Is Proven
    - No proof of control
    - Proof of Last Transaction Control
    - Proof of Full transaction history
- 2WP Designs
  - Single Custodian
  - Multi-sig federation
  - Sidechain
    - Rootstock plan to support sidechains
    - Case study: Sidechain, secondary chain miners have custody, no merged-mining
    - Case study: Sidechain Bitcoin miners have custody, merged-mining
    - Case study: Sidechain, hybrid custody group with insurance through monetary bonds
    - Optimizations, confirmation, contest and post-locking periods
    - Sidechain implementation using Smart-contract in the secondary chain side
    - Sidechain implementation using specific opcodes in the Bitcoin side
    - Sidechain implementation using Turing-complete scripting in the Bitcoin side
  - Drivechain
    - Drivechain implementation today
    - Drivechain implementation using PROP/VOTE Tags
    - Drivechain implementation using only the VOTE Tag
    - Drivechain implementation allowing negative votes
    - Case study: A hybrid drivechain and federation with secoin bonds
    - Case study: A hybrid drivechain and a federation (allowing federation votes)
    - Implementation using reflection and Turing-complete scripting
  - Technical Complexity
  - Security
  - Openness
  - Acceptance
  - Legality
- Conclusion
- References

Introduction

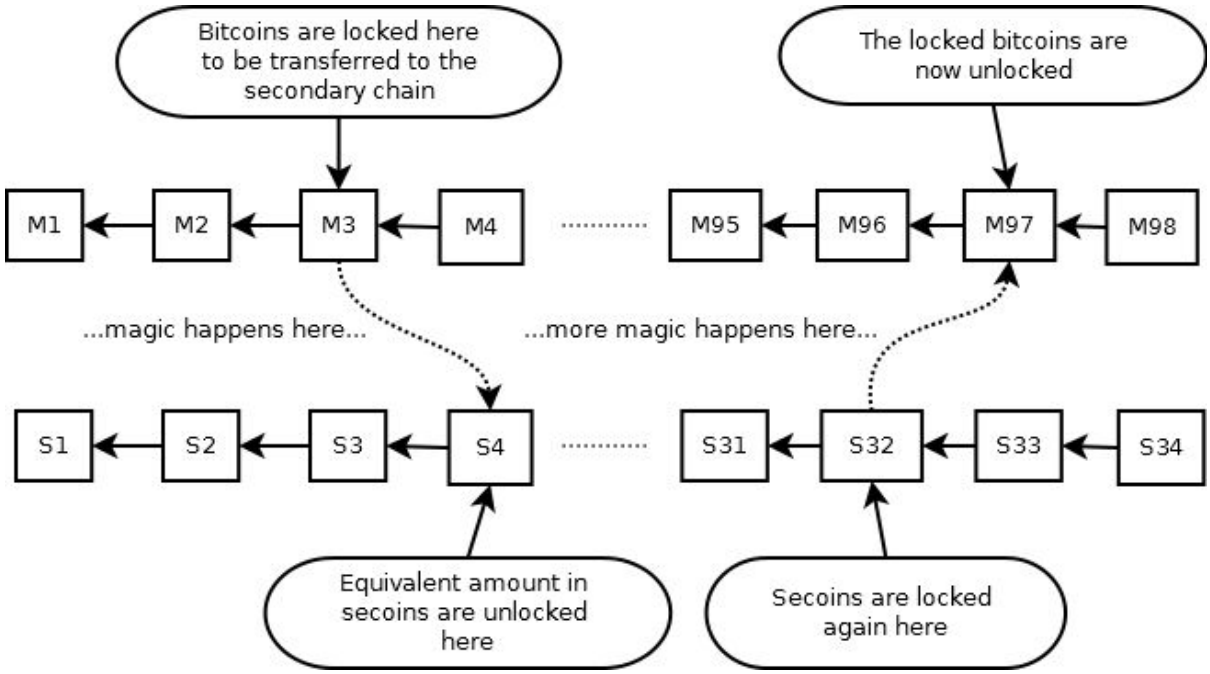
A 2-way peg (2WP) protocol is a protocol that allows transfers of a cryptocurrency from a main blockchain to a second blockchain and vice-versa requiring low third-party trust. This article compares the existing 2WP methods for a blockchain: a Federation, a Sidechain and Drivechain. We also investigate hybrid methods and try to evaluate them to choose which is best suited for the 2WP of the RSK platform on each stage of adoption.

The so-called Sidechain method was created by Blockstream and is based on a peer-to-peer token exchange method created by Sergio Demian Lerner (P2PTradeX) [2]. Drivechain is a newer and simpler method developed by Paul Sztorc from Truthcoin [3].

Along this document we compare pros and cons of each method from different perspectives (technical complexity, security, threat model, community acceptance, openness, legality). We also propose improvements to drivechain and sidechain implementations, which we believe are simpler and more secure.

We present some definitions that will facilitate the explanations along this document. In a 2WP, one of the blockchains is the “main chain” and the other is the “secondary chain”. For our purposes, the “main chain” will always be Bitcoin. To differentiate between bitcoins living in the Bitcoin blockchain from bitcoins living in the secondary chain, we will call the latter “secoins”.

Any method for creating a 2WP requires bitcoins to be locked and secoins to be unlocked or created. The other way around, secoins need to be locked or destroyed and bitcoins unlocked. The distinction between creation and unlocking of secoins and destruction or locking of secoins is mostly irrelevant. Either there are 21M secoins pre-created in a locked state and unlocked when necessary, or there are no secoins pre-created and they are minted on-the-fly when bitcoins are locked. Only in the improbable case that the upper cap of 21M bitcoins is modified by a hard fork is that the distinction makes sense. If secoins are unlocked, instead of created, then the total amount of secoins in existence will be capped to 21M no matter how many bitcoins exist.



The 9 Kinds of Magic

We have researched 9 ways a 2WP can be implemented, in terms of the security of the transferred funds and the threat model, specified by 2 orthogonal criteria:

Who has custody:

- External party or group (Secondary chain miners or stakeholders, a federation)
- Internal party or group (Bitcoin miners, Bitcoin stakeholders)
- A mixture of internal and external parties, in any voting structure.

What proof the custodians require to release funds:

- No proof
- Proof of Last transaction control (also known as SPV control)
- Proof of Full transaction history and last transaction control

Who Has Custody

The custody of locked bitcoins can be either external to the main chain or provided by internal actors of the main chain, or any mixture of them, either by conjunction (both must authorize) or by disjunction (any of them can authorize). There are many possible cases with varied security, and we'll explore the most evident ones.

External party or group has the custody

In a 2WP with an external custody, the funds are ”locked” by sending them to an address of an external party or to an address that can be spent if this external party provides an specific

proof of their authority. Whoever happens to authorize the unlocking of these funds is out of the scope of the main chain and there is no protocol link between the main chain and the external group in custody apart from scripting code in Bitcoin outputs that makes the connection. We distinguish between two possible external custody cases:

- The miners of the secondary blockchain are custodians
- Stakeholders of secondary blockchain are are custodians
- A hybrid where miners of the secondary blockchain are custodians but also have monetary bonds

In the first case scenario (secondary chain miners are in custody), we again make 4 new options:

- The secondary chain is merge-mined
- The secondary chain uses the same PoW function but does not allow merged-mining
- The secondary chain uses a different PoW
- The secondary chain uses a private consensus method (e.g. PBFT)

Miners are custodians, the secondary chain is merge-mined

If the chain is merge-mined the external group is actually an internal group. Thus, this case will be analyzed in the following section.

Miners are custodians, the secondary chain uses the same PoW function but does not allow merged-mining

If the secondary chain uses the same PoW, but not merged-mining, then the hashing power competition between chains goes against the security of both chains. Assuming enough liquidity on both blockchains, and assuming the difficulty adjusting algorithms of both chains prevent high frequency oscillations in hashing difficulty, miners will slowly move to the blockchain having higher revenue, and the opposite blockchain will die with no security. In a nutshell, because of this competition for security providers, there cannot be synergy between both blockchains.

Miners are custodians, the secondary chain uses a different PoW

If the secondary chain uses a different PoW function, it will have to find ways to incentivise mining (pay of the electricity consumption of mining). The only practically proven way to incentivize non-merged-mined PoW mining is by creating a new native token and subsidizing mining by minting tokens. Creating a new speculative token competes with Bitcoin, therefore there will be little support by Bitcoin holders for this secondary blockchain. This would be the case of Bitcoin-denominated tokens created in Ethereum using a sidechain or using “Bonded Escrow Contracts”**.** Although this setting may work, we find that it do not fulfils the interests of the Bitcoin community.

The secondary chain uses a private consensus method

In case the consensus is reached by other means, such as a federation, then the federation can be liable for stealing secoins, so this setup is sound, although more centralized. Proof-of-stake consensus cannot be securely bootstrapped for a secondary blockchain using a 2WP, since the first party to transfer coins to the secondary chain would automatically get the custody of most of the secoins, so the locked bitcoins become easy target for theft if the first party to transfer is dishonest.

Stakeholders of secondary blockchain are are custodians

One interesting case is when the stakeholders of the secondary blockchain are in custody of the secoins using proof-of-stake. This case deserves further analysis.

Hybrid where miners of the secondary chain are custodians but also have monetary bonds

One variant is that secondary blockchain miners must lock a bond in secoins to be able to have custody of the bitcoins. This will be studied in the sidechain section.

Bitcoin miners have the custody

In a 2WP it seems natural that the Bitcoin miners take custody of the funds moved to the peg. The security implication of this setup is different from the scenario where the custody is held by an external party: the trust in the main chain can deteriorate considerably if its own consensus group is shown to be acting maliciously regarding the secondary chain. If more than 50% of miners are merged-mining a secondary chain and they attempt to steal the secondary chain funds, then the confidence they won't to perform a 51% attack on Bitcoin will quickly deteriorate. Bitcoin security against the 51% attack relies on economic incentives. The strongest economic effect that protects Bitcoin from the 51% attack is a mutual-assured-destruction (MAD) property: if a group of miners attempt to perform a 51% attack, everyone, including the miners performing the attack, lose. It is difficult to benefit from a Bitcoin price collapse, as nobody lend bitcoins at the scale of hundreds of millions of dollars. It's very difficult to predict what will happen in the extreme case of a 51% attack, but we can speculate that users will notice the attack attempt and switch to other memory-hard mining PoW function or even users could revert the funds stolen. In the past, we've seen successful last minute collaboration to solve consensus problems. So miners will lose all their hardware investment and probably also their bitcoins used for double-spend. The maturity period of coinbases provides some additional disincentive and strengthens MAD, as the unmatured coins serve as bonds, but these coins are minimal compared to the bonds created by miner's investment in hashing hardware.

If a secondary chain fails (either by being attacked or by critical bugs) the consequences in the main chain are limited, and only affect the funds locked for that specific secondary chain. This is a desired property, and allows the users to extend the functionality of Bitcoin (the unit

of account) with low risk, and without increasing the attack surface. But at the same time the damage control is the achilles heel of a 2WP in many situations. For example, in case of a merge-mined setup where only a minority mines the secondary chain. If this minority attempts to steal the secoins, the minority can be blamed for the attack, and the argument that the majority is honest will still hold. Then, the desired delicate equilibrium for MAD is never reached, so the security of the peg is substantially lower. Only when 51% of Bitcoin total hashing engagement is merge-mining the peg is secure by the MAD property.

Bitcoin stakeholders have the custody

The group in custody of the secoins can be the Bitcoin stakeholders, using a proof-of-stake scheme. The incentives for this group to protect the secoins is unclear, since Bitcoins can be kept anonymously for long periods. It seems that the existence of a small percentage of stolen bitcoins in circulation does not affect Bitcoin price. Only if a huge percentage of bitcoins are stolen, such as the case of MtGox, there was a run to the currency. So the bitcoin stakeholders having custody of secoin funds does not provide enough MAD tension.

What kind of Control Is Proven

There are three levels of proof the custodians can provide, from no control proof to the full history of transactions moving the coins from unlocking to re-locking.

No proof of control

The easiest case to implement a 2WP is when the custody group can decide to unlock locked coins without requiring any authorization from users. As a group, they are trusted. For instance they can conform a federation where the majority of votes is required to unlock funds. Truthcoin can be defined as a 2WP with internal custody and no proof of control, as will be described in a following section.

Proof of Last Transaction Control

The main chain can request the custody group to provide a proof that a transaction that commands secoins to be converted into bitcoins be buried in the secondary blockchain (e.g. has 100 confirmations). This is done using an SPV proof. This requirement restricts when and how much the custody group can unlock. Suppose that the custody group is the consensus group of the secondary chain, and that the secondary chain is proof-of-work mined. To perform an attack, the custody group has to create an alternate branch where secoins are not locked (and therefore stealing both the bitcoins and the secoins). But that has a cost in terms of electricity consumed to build the alternate branch, and under assumptions about the available hardware for lease also requires a minimum amount of

time. Blockstream's sidechain concept is based on Proof of Last Transaction. Proofs of last transaction control are also adequate when the secondary chain was not conceived to support a privileged pegged asset, such as Ethereum, and allows any user to build his own 2WP. However, this proof requires an SPV proof, which even if compacted, can be very costly in terms of transaction size.

Proof of Full transaction history

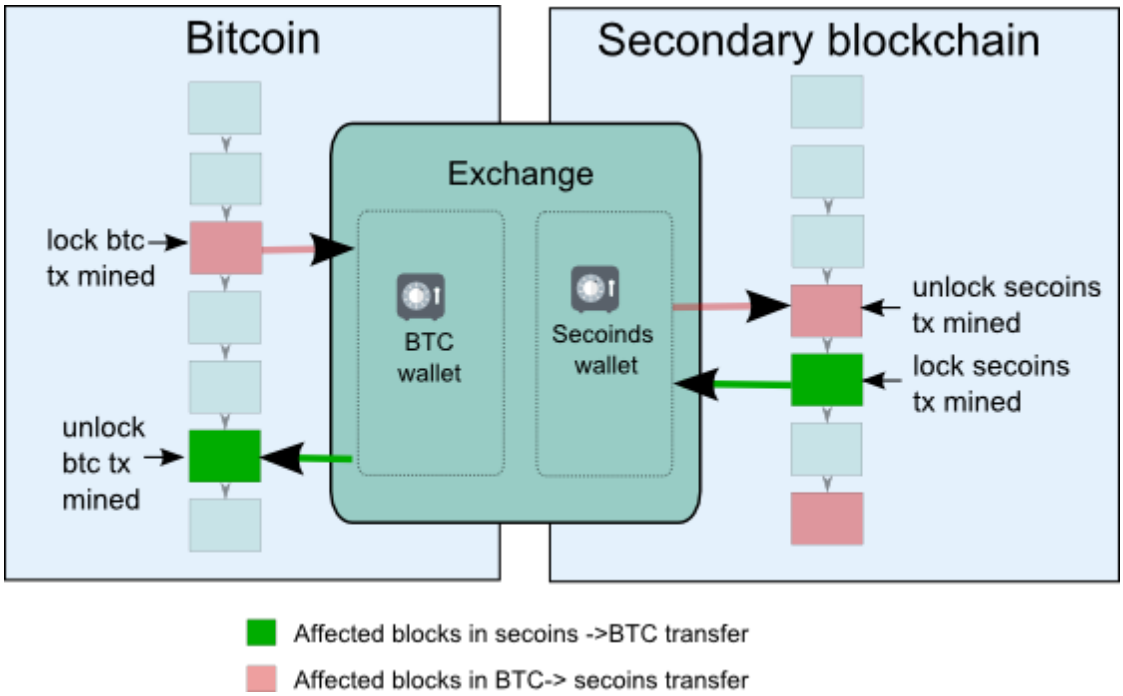
A theoretical variant of the previous method is that the custodian must show a proof of the full chain of transactions from the moment the secoins were unlocked to the moment they are locked again for transfer. The straightforward implementation of this method for a divisible cryptocurrency seems inefficient, if not impossible. The chain degenerates into a tree with of near geometric growth. Soon almost the full secondary blockchain would need to be shown as evidence. Only SNARKS-based approach may reduce the size of the proof, but it will introduce a step of third party trust: the setup. For non-divisible tokens, the approach may work. However the benefit is very limited, as the custody group can try to tap into the transfer chain and perform the same attack.

2WP Designs

One of the properties of existing 2WP solutions is that they try to be symmetric, using the same proof mechanism for one side of the peg and the other. Although this may simplify the design, the best technical and security choice may not be a symmetric design. For instance, the designers of a merge-mined secondary chain that uses Bitcoin as its main cryptocurrency may assume that Bitcoin will always have higher mining engagement than the secondary chain, and therefore rely on SPV proofs for the secondary chain side, but not on SPV proofs for the mainchain side. This is the case of Rootstock.

Single Custodian

The simplest option to implement a 2WP is having a single custodian (e.g. and cryptocurrency exchange) holding custody of the locked bitcoins and holding custody of unlocked equivalent tokens. The exchange would manually enforce the promise of locking bitcoins before unlocking secondary tokens either manually or by means of a protocol executed in software. This setup is depicted here:



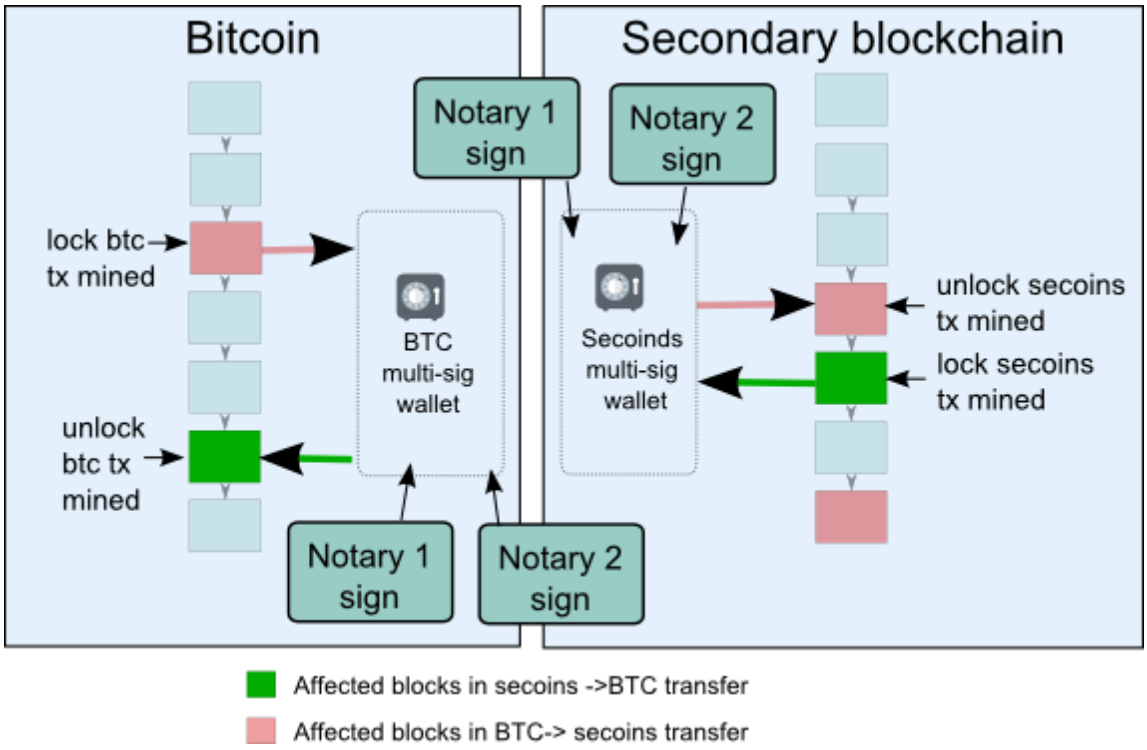
The drawbacks are having a single point of failure and complete centralization in the control of the funds.

Multi-sig federation

A better option to implement a 2WP is having an group of notaries control of a multi-signature, where the majority of them has to approve the unlock of funds. This setup better than having a single controller of the funds, but still centralizes control. To achieve true decentralization, the notaries should be carefully selected to have at least the following properties:

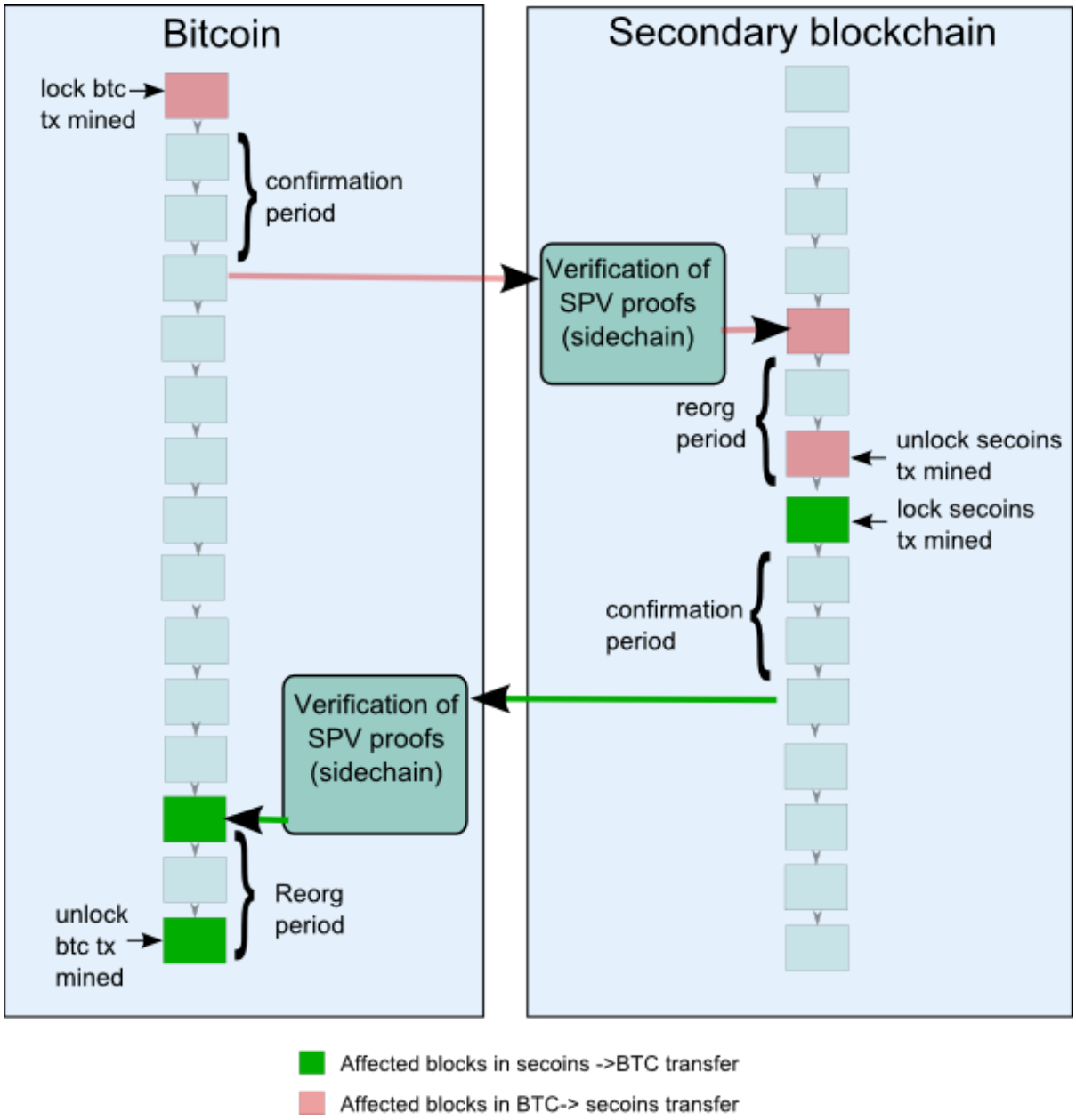
- the number of notaries should not be low (e.g. at least 10).
- the number of notaries should not be too high, so that users can verify the authenticity and honesty of the notaries (e.g. below 30).
- Notaries should be distributed across different legal jurisdictions and nations, to prevent state attacks, coercion and censorship.
- Notaries should be geographical distributed to prevent failure of the infrastructure on natural disasters.
- Notaries should be renown.
- Notaries should not be controlled (or dependant on) a lower number of entities. For example, notaries cannot be different branches of the same bank.
- Notaries should be able to achieve and maintain a specified level of security through physical and logical protections, together with required security procedures.

This setup is depicted here:



Sidechain

The sidechain method is a symmetric 2WP based on each chain understanding how the other chain creates consensus to lock or unlock coins for cross-transfer, and exchange short proofs of last transaction control for locking/unlocking. To easily explain this concept first suppose that the secondary chain has “settlement finality” and the consensus is reached when a fixed number of parties (the consensus group) sign the block containing the cross-chain transfer transaction (e.g. by using a Byzantine fault tolerant consensus). In this example clearly the mainchain can be provided the signed block as the payload of a main chain transaction, together with an SPV proof, and that’s enough for the main chain to unlock the equivalent amount bitcoins and send them to where the payload specifies. The difference from this approach and a multi-sig of notaries receiving the funds for custody is that the consensus group need not to be aware of the 2WP: by signing each block the secondary blockchain consensus group members are providing enough proof for the main chain to unlock the funds. The secondary chain consensus group has full power to unlock the funds in the main chain and send them to whatever address they want (by creating an alternate signed branch, for example). If the secondary consensus method is proof-of-work, then an SPV proof containing confirmation headers must must be used, and the primary blockchain requires more complex protocol to prevent being presented a fake SPV proof by choosing the proof with higher accumulated work from all candidates presented. The setup of a sidechain is depicted here:



Rootstock plan to support sidechains

Rootstock aims to provide two ways for building and verifying SPV proofs. The 2WP smart contract in Rootstock generates two different indicators announcing the wish to unlock bitcoins. The contract emits a log entry containing the bitcoin unlock transaction and also Rootstock can force the miner to add the hash of that transaction to a protected virtual Bitcoin-like Merkle tree referred in a virtual Bitcoin-like header, so Bitcoin can easily understand it, even if Rootstock internal transaction format and header is completely different from Bitcoin format. The Rootstock block is valid if this virtual Merkle tree contains the element which the log announces. The SPV proof validation code can check the the spending transaction equals the transaction announced in this log entry (if it is smart enough), or navigate the virtual Merkle tree and find it there (if it only accepts Bitcoin-like secondary chains). The choice of the method Rootstock will ultimately use will depend on

the way the 2way is finally supported in Bitcoin. If the support is generic enough, then the virtual Bitcoin-header will not be necessary.

Case study: Sidechain, secondary chain miners have custody, no merged-mining

One way to provide higher security compared to using a multi-sig address is that an external custody group has a high stake in the secondary chain: for example, if the custody group is exactly the the secondary chain miners. For the custody group to have that high stake, we must assume that collecting transactions fees in the secondary chain is a good business in itself. Suppose that all Bitcoins in existence are transferred to a sidechain. Bitcoin fees are not proportional to the money exchanged, and taking into account the measured velocity of Bitcoin, and assuming the current transaction fees stay unchanged, it would take at least 1000 years for the miners to earn in fees the same amount of the total of Bitcoins minted. Even if one assume that the sidechain will only hold the number of bitcoins active in the last 24 hours (about 4M) or the number of bitcoins transacted in the last 24 hours (about 400K), the business of collecting fees is nothing compared to the “business” of stealing secoins (the former only achieves the profits of the later in 30 years or more).
So the conclusion is that either the secondary chain has proportional transaction fees, such as 0.5% or the custody group is exactly the Bitcoin miners, and more than 51% of the Bitcoin miners are engaged in merged-mining the secondary chain. In a smart-contract platform such as Rootstock or Ethereum achieving a proportional transaction fee is near impossible, since miners are not aware of the meaning of each message, and it’s easy to hide transfers in arbitrary messages sent to a contract that takes custody of coins. It’s also difficult in blockchains that cryptographically hide the payment amount, such a zCash. The conclusion is that the secondary chain miners are not good candidates of being in custody of the bitcoins, if there is no merge-mining.

Case study: Sidechain Bitcoin miners have custody, merged-mining

Let’s now assume that the secondary chain is based on proof-of-work and it is merged-mined, as is the case in Rootstock. Let’s also assume that 90% of the miners are merged-mining. Now the network can detect if a large subset of miners are attacking the secondary chain because the Bitcoin blocks will show there are two or more competing chains being created. Also they can detect a sudden drop of support for merged-mining. Since an engagement below 50% is dangerous, users should be able to detect if sidechain miner engagement is below 50% compared to Bitcoin engagement, or Bitcoin all-time high. If so, the sidechain node application must warn the user to immediately move the secoins back into the Bitcoin blockchain, or move the funds in custody to another custody method, such as a federation.
In terms of security, if the secoins miner engagement is below 50%, the majority of secoins miners can try to steal secoins this way: they acquire secoins, and transfer them back to bitcoin addresses owned by them using the peg, then they create a new blockchain branch

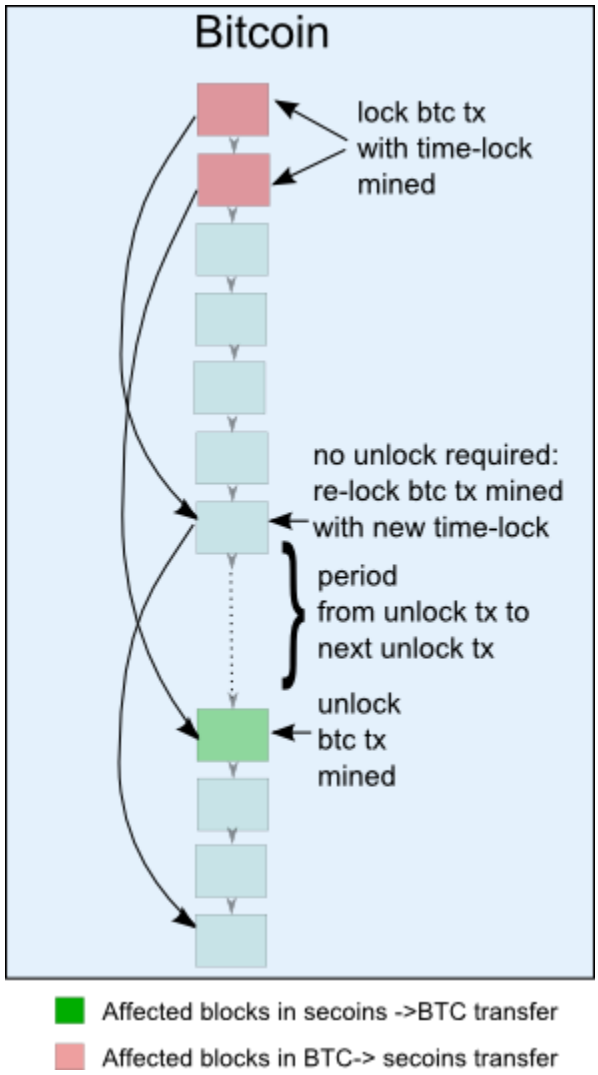
where the same secoins are not sent to the peg, but to an crypto-exchange where they exchange the secoins for bitcoins.

Case study: Sidechain, hybrid custody group with insurance through monetary bonds

A subgroup of secoins stakeholders (“bonders” from now on) can take part in the custody group by holding insurance bonds that are locked for some time until a peg transaction settles. The bitcoins that are locked must be also time-locked using CLTV so that the rate of unlocking cannot be higher than what the bonds allow. In other words, a secondary chains smart-contract only allow to unblock secoins if the bitcoins are time-locked at the desired maximum unlock rate. The result is an hybrid model with the following properties:

- Unlocking bitcoins require the approval of the bonders and the secondary chain miners, by means of an multi-sig (bonders) and a SPV proof to an unlock transaction (miners).
- The unlock transaction is chosen by a smart-contract that hold the bonds of the bonders. The unlock transaction is chosen carefully to unlock bitcoins whose time-lock has expired.
- Both the bonders and secondary chain miners must periodically approve spending the locked bitcoins to new lock scripts with increasing CLTV time-locks for coins whose time-locking time have elapsed but were not requested to be unlocked (re-locking).
- To steal the bitcoins in custody, the bonders require the help of 51% of the miners, to build the longest SPV proof. Even in this case, they can only steal at a pre-established rate.

The following diagram depicts how lock transactions are also time-locked:



This hybrid model increases the MAD tension sacrificing the trade volume but without sacrificing the time a single cross-chain transfer takes. The bitcoin locking scriptPub will look as the concatenation between a federation using multi-signatures and a sidechain validation script.

```
<expiry time> CHECKLOCKTIMEVERIFY DROP

<N> <N/2+1> <FedPubKey1> … <FedPubKeyN> CHECKMULTISIGVERIFY
<spv-chain-verification-code>
```

In the case of Rootstock, the spv-chain verification and cross-check is handled in a single operation, because the 2WP smart contract emits a messages containing the exact copy of the Bitcoin spending transaction of the locked bitcoins.

This method has the drawback that the federation can extort the secoin holders and refuse to unlock bitcoins. However, as they have secoin bonds, they enter a deadlock where they cannot recover the bonds nor can users transfer back bitcoins. The disincentive to this attack

would be much more legal (the federation is composed by known members) rather than economical (the bonds could be substantially lower than the secoins unlocked).
Also this method has the drawback that the price of a secoin may drift from the 1:1 desired exchange-rate because the rate limited peg reduces the chances secoins can turned into bitcoins fast in case the secondary chain infrastructure (exchanges, miners, nodes) fails.

Optimizations, confirmation, contest and post-locking periods

Standard Proof-of-work consensus does not have settlement finality, and the probability a best chain becomes immutable is based on an economic model, not on a cryptographic hardness assumption, and of course not on a mathematical truth. Therefore the proof of last transaction control must include a number of confirmation headers for cheating to be expensive and evident by a decrease in apparent network hashing power. The contest period gives us an new opportunity: we can turn the proof into an interactive proof and reduce the amount of information as proof payload. The custody group can present a short proof (providing proof for a lower proof-of-work, or a proof of work with lower confidence factor) and expect anyone to contest it. If no one contest it, it is accepted by the main blockchain. The blockchain should not accept proofs of too low proof-of-work, because in that case mainchain users cannot differentiate between a contest period with a single authentic proof having low proof-of-work and proof of a counterfeit transaction followed by censorship to publish any contested proof by the miners. It turns out that using a previous block skip-list embedded in the chain header, and a challenge-response protocol for querying past headers turned non-interactive using the Fiat-Shamir heuristic a miner can prove 90% of the work of a 1000 block long chain by providing only 20 headers (together with the merge-mining proof). To improve the security of the system, any proof provided to unlock bitcoins must be followed by a post-lock period where nothing happens, just to make sure that if the miners try to steal, they cannot profit from the attack (MAD property). The conjunction of confirmation, contest and post-lock periods is what makes sidechain a slow system to move bitcoins between blockchains.

Sidechain implementation using Smart-contract in the secondary chain side

Rootstock currently uses an asymmetric 2WP: the secondary chain can unlock secoins when a privileged native smart contract (the Bridge) receives an SPV proof in a message. The Bridge is also manages the creation of the Bitcoin transaction to release unlock bitcoins by creating a proposal and expecting signatures from federators. Also it manages adding and removing federators.

Sidechain implementation using specific opcodes in the Bitcoin side

The logic to validate a foreign blockchain consensus is difficult to generalize in a way a single opcode can cover all cases. In cases where the secondary blockchain is a fork of Bitcoin, the gamut is narrower. Elements uses OP_WITHDRAWPROOFVERIFY and OP_REORGPROOFVERIFY opcodes to understand Bitcoin consensus and proof-of-work, and similar opcodes could be used in Bitcoin to understand a secondary chain consensus. However this opcodes do not easily generalize for Ethereum blockchain. Rootstock has the capability to emulate a Bitcoin header and be compatible with such opcodes, but this is far the generic solution Bitcoin requires.

Sidechain implementation using Turing-complete scripting in the Bitcoin side

These are the operations Bitcoin scripting should be able to perform to evaluate a generic foreign blockchain consensus:

- Evaluate an SPV proof or a compact SPV proof
- Manipulate and extract information from a byte array: OP_SUBSTR, OP_CAT
- Compute proof of work: OP_HASHxxx or all arithmetic and logic and table lookup operations used in the construction of hash function.
- Compute PoW difficulty: OP_DIV, OP_MULT
- Postpone a change in state until a certain block timestamp or height.
- Persist information to be able to switch from one state to another state. This can be done using reflection by hashing the scriptPub of the first output of the spending transaction and verifying that script hash corresponds to the script hash of the next state script. The most generic opcodes to allow it are: OP_PUSH_SPENDING_TX, OP_PUSH_SPENDING_INPUT_INDEX. The transaction parsing would be done in the script. Another option is to be able to directly access spending transaction parts: OP_PUSH_SCRIPTSIG, OP_PUSH_SPENDING_TX_OUTPUT, OP_PUSH_SPENDING_TX_OUTPUT_COUNT.
- Persist the amount. This requires to be able to parse the spent transaction output (OP_PUSH_PREVTX, OP_PUSH_PREVOUT_INDEX). Another simpler approach is to be able to push the input amount using OP_PUSH_AMOUNT.

This is a example script pseudo-code to perform the withrawalproof and reorgproof:

1. Assume stack contains: <spv-proof>
2. push in altstack the work-of-best-spv-branch (or zero initially)
3. push in altstack the block height when last spv-proof was received (or zero initially)
4. Compute the work of the spv-branch passed as argument.
5. If the work is below the minimum work required for the SPV proof, abort.
6. If first element of altstack is zero, removed the two elements from the altstack, push the spv-proof and push the block-height in the altstack, then goto step 11.
7. Parse the spending transaction, extract the nLockTime.

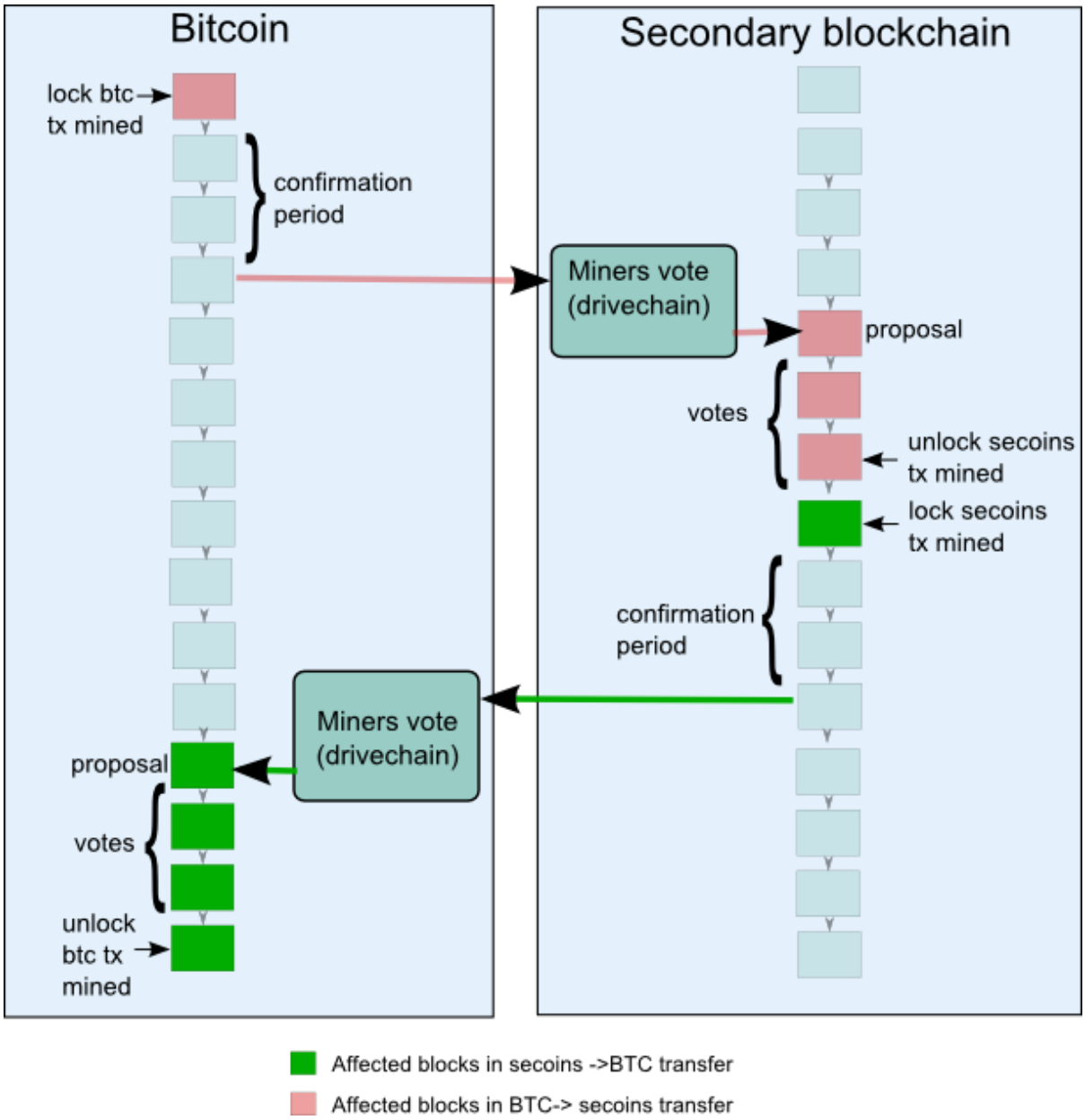
8. Use CHECKLOCKTIMEVERIFY to make sure the nLockTime is higher than the block-height value existent in the altstack.
9. Verify that the amount of work computes is higher than the amount of work stored in the first slot of the altstack. If not, abort.
10. Remove the two elements of the altstack. Store the nLockTime and the current spv-proof amount of work in the altstack.
11. Read the first output of the spending transaction. Verify that the amount is equal to the amount received in this input, and abort if not equal. The remaining inputs and outputs can be used for fees.
12. Get the spending transaction input. Parse the scriptsig. Check that the two first instructions corresponds to push ops on the altstack the contents of the current altstack. If not abort.
13. Verify that the hash digest of the remaining of the scriptsig corresponds to the hash digest of this script tail (the two initial push operations are excluded)

To summarize, the following opcodes need to be added to Bitcoin:

- OP_SUBSTR
- OP_JUMP
- OP_CONDITIONAL_JUMP
- OP_PUSH_SCRIPTSIG
- OP_PUSH_AMOUNT
- OP_PUSH_SPENDING_TX_OUTPUT

Drivechain

A drivechain is a 2WP with no proof of control and provided with internal custody. In a nutshell, a drivechain gives custody of the locked btc to the Bitcoins miners, and allows Bitcoin miners to vote when to unlock bitcoins and where to send them. The miners vote using the bitcoin blockchain, and votes are cast in some part of the block (e.g. the coinbase). If the proportion of miners who participate in voting is under 50%, then the remaining ones can easily cheat by voting whatever transfer that let them steal the bitcoins. If the miners can vote quickly to unlock the locked coins and spend them, they can cash out (in other cryptocurrency or in fiat) before the community reacts and finds out the majority of Bitcoin miners are dishonest. Therefore, as in a sidechain, post-lock period must also exist as MAD disincentive. The following diagram depicts a drivechain:



Drivechain implementation today

In the simplest possible implementation, the bitcoins transferred to a drivechain are sent to a non-standard anyone-can-spend output. Therefore, only miners will be able to generate such transactions. The output should also specify the secondary chain id and destination address. As voting is done by miners, a soft-forking rule that only affects miners would stipulate when these outputs can be spent, based on the previous votes. The non-standard nature of these transactions would prevent miners not aware of the soft-forking rule to mine invalid transactions send by malicious users that spend the outputs. Even if this could be implemented by the miners without any support from the Bitcoin users, we believe than all consensus rules the miners adhere should be public, and that is a key part of the open nature of Bitcoin. Therefore, we propose several implementations using an explicit opcodes.

Drivechain implementation using PROP/VOTE Tags

A drivechain can be implemented in several ways and we propose one of such possibilities, which is not the original one presented by Truthcoin. Miners that want to propose a candidate embed the message “PROP”.

New coinbase tags:

- PROP: <secondary-chain-id> <poll-id> <proposed-spend-tx-hash>
- VOTE: <num-votes> <proposal-id(1)> .. <proposal-id(num-votes)>

The proposed-spend-tx-hash is the hash of the transaction that is proposed to be included in a block. Note that the miner that proposes the transaction may have to pay higher fees because of the block space consumed by the proposal (e.g. 70 additional bytes).

This is low, but the secondary chain protocol could compensate this miner allowing that the spending transaction includes an output to a proponent miner’s address, with a specific pre-established fee. The proposal id can be the hash of the proposal message, or a monotone increasing proposal counter. The poll-id is a blob that identifies the current poll, in the secondary chain parallel polls for parallel unlocking. It is not used by the Bitcoin protocol, but it can help miners track the poll. There can be many active proposals for the same drivechain, and a miner can vote for many of them simultaneously. There is no way to vote against a proposal. This is to prevent a rogue miner from creating too many proposals and forcing the remaining miners to vote against all of them, consuming block space. The miners can vote for any proposal by embedding the message “VOTE”, followed by proposal id, in the coinbase.
The voting period has a hardcoded maximal value (max-voting-period). A practical value for max-voting-period is 720 (approximately 5 days). Also there is a hardcoded limit on when a unlock transaction can be spent after the corresponding voting is over (max-liveness-period). A practical value for max-liveness-period is 864 (6 days).

The opcode OP_CHECK_POSITIVE_VOTES_VERIFY is added. This opcode receives the stack arguments:

- secondary-chain-id
- voting-period (in blocks)
- liveness-period (in blocks)
- min-number-of-positive-votes

This opcode computes the most voted candidate. If a blockchain votes more than once for the same poll id, only one vote is counted. The opcode invalidates the transaction if:

1. The liveness period is greater than the maximum liveness period (liveness-period > max-liveness-period)

2. The voting period is greater than the maximum voting period (voting-period > max-voting-period)
3. The transaction specified by the proposal hash does not match exactly the spending transaction. If there is a match, then the remaining computations are related to the
4. The voting period has not passed since the proposal was made
5. Too little positive votes (positive-votes < min-number-of-positive-votes) from the time the poll was created (poll-start) for voting-period blocks.
6. Too old poll (current-height > poll-start+liveness-period)

After voting-period-in-blocks blocks, no new positive vote is counted, but the poll results can still be used until the liveness-period ends.

The liveness period allows Bitcoin to forget about polls that have been abandoned and prevent them to consume ledger state space. Also the protocol should limit the maximum value for liveness, to allow light clients and Bitcoin Core in prune mode to validate the poll. A maximum value of 1440 (approximately 10 days) is suggested.
To be implemented as a soft-fork, the OP_CHECK_POSITIVE_VOTES_VERIFY (CPVF) should replace an OP_NOP opcode.

Example scriptPub: `"ROOTSTOCK" 7 20 4 OP_CHECK_POSITIVE_VOTES_VERIFY`

Example blockchain that can spend the scriptPub:

Block	Coinbase tags	Prop 1	Prop 2
1	PROP: “ROOTSTOCK” 1 0x97f5cb1….86a8b8ec	1	0
2	{no votes}	1	0
3	PROP: “ROOTSTOCK” 2 0x4d5aef….16116cab3	1	1
4	VOTE: 0x01 0x01	2	1
5	VOTE: 0x01 0x01	3	1
6	VOTE: 0x01 0x01 0x02	4	2
7	Contains the transaction specified by hash 0x97f5cb1….86a8b8ec	4	2

Drivechain implementation using only the VOTE Tag

A drivechain can be implemented using only the VOTE tag, and this has both benefits drawbacks. Miners that want to propose or vote for a candidate embedding the tag “VOTE”.

New coinbase tag:

- VOTE <proposed-spend-tx-hash>

One of the benefits of using a single idempotent tag is that the merge-mining module does not need to know the state of the bitcoin blockchain. It can emit votes until the secondary chain stops requesting it. An on-chain message containing the SPV proof that the bitcoin unlock transaction has been confirmed can be sent to the peg smart-contract so secondary chain can inform the voting has concluded. The drawback is that the VOTE tag requires much more space and may be required to be embedded in coinbases even after the proposed spend transaction has already been confirmed, until the secondary chain is notified of such confirmations.

Block	Coinbase tags	Prop 1	Prop 2
1	VOTE: 0x97f5cb1….86a8b8ec	1	0
2	{no votes}	1	0
3	VOTE: 0x4d5aef….16116cab3	1	1
4	VOTE: 0x97f5cb1….86a8b8ec	2	1
5	VOTE: 0x97f5cb1….86a8b8ec	3	1
7	VOTE: 0x97f5cb1….86a8b8ec VOTE: 0x4d5aef….16116cab3	4	2
8	Contains the transaction specified by hash 0x97f5cb1….86a8b8ec	4	2

Drivechain implementation allowing negative votes

The voting system described in the previous sections only count positive votes. This is to prevent rogue miners from creating too many malicious proposals and force every other miner to vote against. However, the CPVF opcodes proposed cannot tolerate correctly a decrease in miner engagement, since the amount of positive votes required to unlock the bitcoins is chosen when they are locked, which can happen any amount of time before unlocking them. Therefore the CPVF opcode is only well suited to be used in conjunction with a federation in a special script so that missing hashing power can be dynamically replaced by federation votes, as we'll see in the following sections. If no federation will provide the missing votes, it is desirable that the poll allows both positive and negative votes to allow unlocking if the mining engagement drops. We assume that if a miner is monitoring a certain secondary chain, it won't ignore a proposal for it: either it will vote positively or negatively. Therefore we can assign a boolean value (1 bit) to every proposal for a drivechain. A positive vote is represented by 1. A maximum voting period

(max-voting-period) is hard coded, so that any proposal consumes 1 bit for miners that care about that secondary chain until a maximum of max-voting-period blocks.

New coinbase tags:

- PROP <secondary-chain-id> <poll-id> <prop-liveness-period> <proposed-spend-tx-hash>
- VOTE <secondary-chain-id> <vote-bin-value>

The new PROP tag adds the argument prop-liveness-period. This argument must be exactly equal to the highest liveness-period specified in the proposed-spend-tx outputs. If prop-liveness-period is higher than max-liveness-period, the proposal is invalid and no vote is counted. The new proposed VOTE tag has completely different arguments. The vote-bin-value is a binary value representing a big integer where each bit maps to a proposal. Proposals whose voting period have elapsed (as specified by prop-voting-period) are not counted in the bit vector, and the vector is shifted accordingly. As opposed to the previous case, the new PROP does not vote for the proposal. The miner that proposes the poll must therefore vote in the same coinbase using VOTE. We propose a new opcode OP_CHECK_VOTES_VERIFY (CVV) that count both positive and negative votes.
This opcode receives the stack arguments:

- secondary-chain-id
- voting-period (in blocks)
- liveness-period (in blocks)
- min-number-of-positive-votes
- max-number-of-negative-votes
- positive-votes-difference

This opcode takes the spending transaction, computes its hash, and searches for the corresponding proposal up to liveness-period of past blocks. If the same proposal is re-proposed, the first proposal counts and the following proposals are considered invalid. The CVV opcode computes all votes (positive-votes and negative-votes) and invalidates the transaction if any of the properties 1-6 described in the preview section hold, and also in the following cases:

1. Too many negative votes (negative-votes > max-negative-votes)
2. If the difference between positive votes and negative votes is lower than the given value (positive-votes - negative-votes < positive-votes-difference)

After voting-period-in-blocks blocks, no new positive vote is counted, but the poll results can still be used until the liveness-period ends.

Example scriptPub: `"RSK" 7 20 1 10 1 OP_CHECK_VOTES_VERIFY`

Example blockchain that can spend the scriptPub:
(Proposals show positive/negative votes)

Block	Coinbase tags	Prop 1	Prop 2
1	PROP: “RSK” 1 0x97f5cb1….86a8b8ec VOTE: 0x01	1/0	0/0
2	PROP: “RSK” 2 0x4d5aef….16116cab3 VOTE: 0x02	1/1	1/0
3	VOTE: “RSK” 0x01	2/1	1/1
4	{no votes}	2/1	1/1
5	VOTE: “RSK” 0x01	3/1	1/2
7	VOTE: “RSK” 0x03	4/1	2/2
8	{no votes}	4/1	2/2
10	Contains the transaction specified by hash 0x97f5cb1….86a8b8ec	4/1	2/2

Case study: A hybrid drivechain and federation with secoin bonds

Similar to the hybrid model proposed for sidechains, a hybrid model can created with a drivechain. In this setup the miners will vote on the unlock transaction which has already been multi-signed by the federation (bonders). This is implemented with a smart contract on the secondary chain that proposes a bitcoin unlock transaction, and collects federator member’s signatures, and emits the final signed transaction only if the signers have enough bonds. Example scriptPub:

```
“ROOTSTOCK” 144 288 10 1 144 OP_CHECK_VOTES_VERIFY

<pubkey(0)> … <pubkey(N-1)>

<N> <M> OP_CHECKMULTISIGVERIFY
```

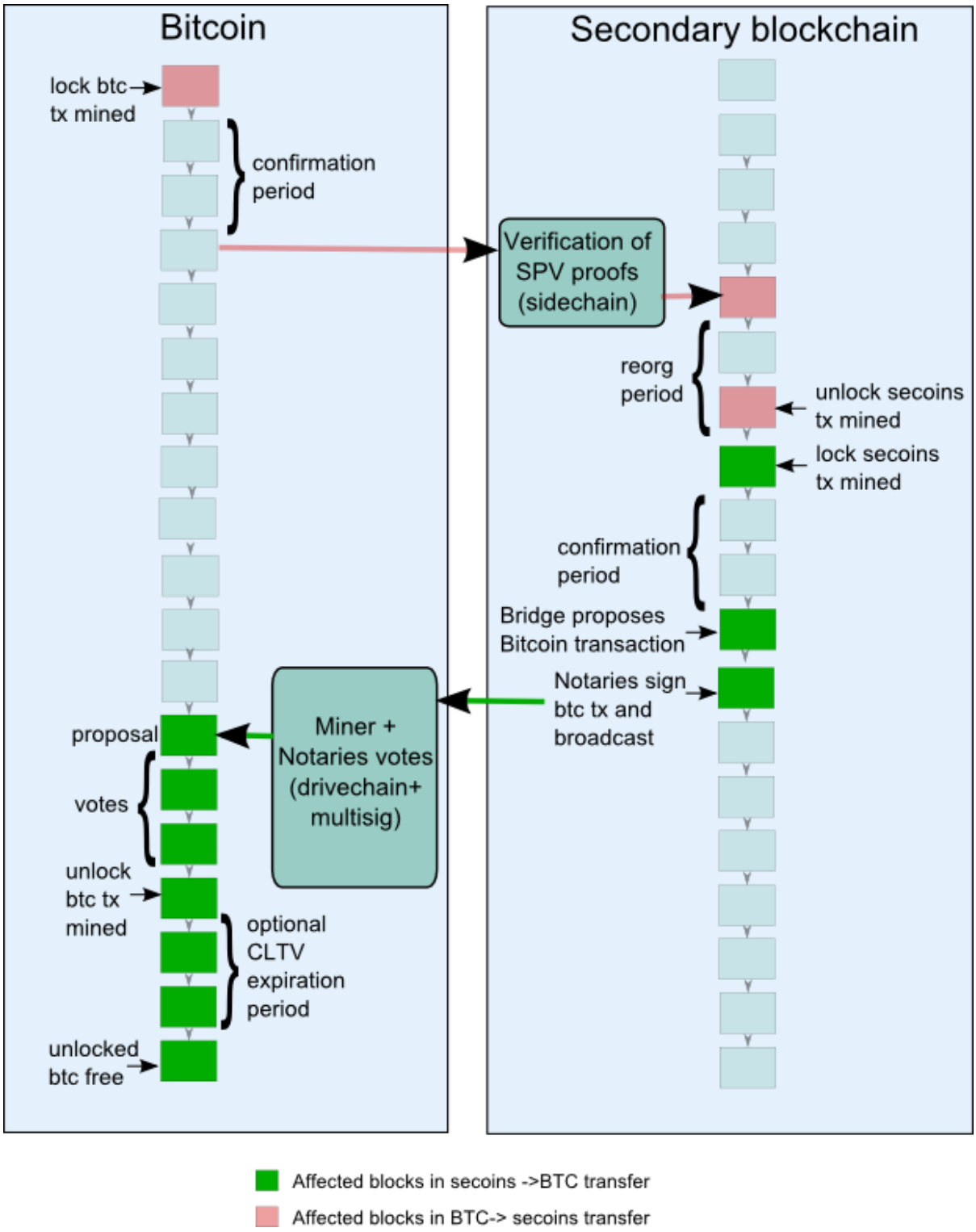
Case study: A hybrid drivechain and a federation (allowing federation votes)

In this setup, only positive miner votes are counted, but missing positive miner votes can be fulfilled by federation member votes. A new opcode OP_CHECK_VOTES_MULTISIG_VERIFY (CVMV) is created, with the following arguments:

- provided-signatures (M) (provided by scriptSig)
- sig(0) … sig(M-1) (provided by scriptSig)
- number-of-public-keys (N)

- pubkey(0) … pubkey(N-1)
- signature-votes
- secondary-chain-id
- voting-period (in blocks)
- liveness-period (in blocks)
- min-number-of-positive-votes
- max-number-of-negative-votes
- positive-votes-difference
- min-number-of-sigs

The argument number-of-public-keys is similar to the CHECKMULTISIG opcode. The argument present-signatures is the number of signatures that follows in the scriptSig. Signature-votes is the number of votes that each signature adds. For example, if signature-votes is 4, then each signature represents the votes of 4 blocks. The min-number-of-sigs is the minimum number of signatures that must be present. It allows the federation to have veto power for any proposal presented by the majority of the miners. The enormous benefit of CVMV is that it allows to bootstrap a merged-mining cryptocurrency from having no merge-mining engagement (all votes must be provided by a federation) to a state where there is a high merge-mining engagement and the federation does not have any control of the funds. There is no point in allowing the federation members to vote against a proposal. This is because the 51% of the miners can censor such negative vote, so security is not increased. If min-number-of-signs is set to zero, 51% of the miners have full control of the locked funds and still can steal the funds, but the MAD tension is maximum. This hybrid method is depicted in the following diagram:



The transactions that unlock the btc can be created with a CLTV time-lock to increase the MAD tension even more.

Implementation using reflection and Turing-complete scripting

The Bitcoin scripting system needs new opcodes and a new dynamic storage data structure that allows easy lookup and computation of votes, as the altstack does not serve this purpose well. This data structure can be a dictionary. Two new opcodes (OP_STORE and OP_RETRIVE) are used to move data from the stack to the dictionary and the other way round. To count votes, coinbase fields of the last N blocks must be examined to detect voting tags, and parse them. A new opcode OP_PUSH_BLOCK_COINBASE pops from the stack the number of blocks to lookup (from the current block backwards) and pushes the coinbase field. The opcodes OP_SUBSTR and OP_CAT are used to search for the tag. A new opcode OP_PUSH_SPENDING_TX is used to push the spending transaction and compare it to the hash of a transaction specified in the candidate tag.

The following opcodes should be added to Bitcoin:

- OP_PUSH_BLOCK_COINBASE
- OP_SUBSTR
- OP_CAT
- OP_PUSH_AMOUNT
- OP_PUSH_SPENDING_TX
- OP_STORE
- OP_RETRIEVE
- OP_JUMP
- OP_CONDITIONAL_JUMP

Comparison

A fair comparison between a sidechain and drivechain must span a number of orthogonal views, since socio-political, business and economic incentives are at stake. Also the best choice for a certain 2WP can be a mixture of methods, such as using a sidechain in one direction of the peg, and a drivechain in the opposite direction.

Technical Complexity

A sidechain requires blockchains to be able to evaluate consensus rules from foreign blockchains and therefore are much easier to implement when the blockchain has a Turing-complete scripting language or VM. If it does not, as in Bitcoin, then implementing it requires a burden of critical consensus code, which may end not being very generic. However some Bitcoin developers are working on a roadmap to add such extension to Bitcoin, and RSK Labs supports this effort: the project Bit2.

On the other side, drivechain requires Bitcoin miners to run full nodes of the secondary chain (or at least special gateways), so their participation is a prerequisite. This limits the chances a non-merge-mined drivechain created by small community gets accepted by the miners. On

the contrary, a merge-mined drivechain would provide some incentives though secondary fees. In most cases, secondary chains supported by companies that are able to establish long term contracts with Bitcoin miners will have higher chance of success. However, every drivechain that adds features to Bitcoin benefits all the Bitcoin community, so this distinction is not decisive.

Security

To compare the security of a sidechain and a drivechain, we assume both secondary chains are using the same merge-mining system with Bitcoin. If the custody group comprises the miners, then both methods, drivechain and sidechain, provide equal security. In drivechain, if a subset of the miners try to steal secoins, they must vote contrary to the authentic secondary best chain, therefore as soon as the voting begins, users running nodes of both primary and secondary blockchains can detect the attempt to steal. However users in the main chain do not see anything special. A special plug-in module for Bitcoin Core could detect contested votings where there is a near 50/50 split between two candidates. Notifications of cheating attempts increases the MAD effect. If a split is detected, individual miners participating in mining pools can refrain from participating in dishonest mining pools, and soon the voting will slide towards the authentic branch. In a sidechain, mainchain users can also use a plug-in module to detect that a reorganization proof is presented, or that the the SPV proof presented corresponds to merge-mined headers whose Bitcoin parent blocks are not present in the Bitcoin blockchain. Most important, the unlock contract can test against this property and reject proofs whose Bitcoin header parents referred by secondary chain headers (or a large subset of them) do not belong to the Bitcoin best chain (this functionality may need additional opcodes from blockchain introspection). Also nodes can detect a secondary chain split by a small mainchain node plugin-module that follows the chains of secondary chain headers embedded in Bitcoin headers when the secondary chain headers contain a skip-list of previous block hashes. To summarize, both drivechain and sidechain methods could allow Bitcoin nodes to detect contending unlocks in advance using simple plug-in modules.
In conclusion, using a sidechain to support unlocking bitcoins when the secondary blockchain is merged-mined is discouraged, as sidechains are conceptually and programmatically more complex, they require unlock transactions that pay higher fees, and they do not provide better security than a drivechain.

Also when the custody group are the merged-miners, we don’t see any crucial difference in terms of security between a drivechain and a sidechain during the bootstrapping period, when there is low engagement of miners in the merge-mining process. Only using a dynamic hybrid federation/drivechain method brings more security during bootstrapping. In this case, the security is higher. If the miners and/or federation must also hold secoin bonds and unlocking bitcoins is rate limited by CLTV, then the security is even higher. In hybrid models, both the federation and the miners need to collude to steal the bitcoins in custody. The drawback is that the federation members can refuse to sign the transaction and therefore extort the network, but this is highly improbable, as federation members have known

identities, reputation and legal liabilities. Rate limiting the unlocking of bitcoins using CLTV in scripts increases the security, but can affect the 1:1 peg rate, because it may reduce liquidity.

Openness

A drivechain requires active participation of the miners for each specific drivechain while a sidechain does not require direct engagement. To support a drivechain Bitcoin requires a soft-fork. It seems difficult to achieve consensus on a soft-fork that will benefit only the drivechain creators who have the resources to convince the miners to join. However, as every properly secured sidechain that adds capabilities to Bitcoin provides benefits to all Bitcoin holders, once a the project shows good quality code and gains interest from the Bitcoin community and Bitcoin miners, positive opinions can push the required soft-fork.

Acceptance

The sidechain concept had a widespread acceptance since Blockstream published its foundational paper. Contrary, drivechain design did not popularize. However we believe drivechains have greater potential for the Bitcoin side of the 2WP for the simplicity of the model, and it will be supported by the community.

Legality

(This section presents only some ideas of our own which have not yet been validated by a competent lawyer. Therefore the contents of this section should not be taken as legal advice)

Two procedures in a 2WP can be a target of regulation. One is the unlocking of secondary currency, which could be interpreted by regulators as the creation of currency, instead of a transfer of currency. A superficial analysis may give this impression if the currency on the secondary chain is pre-created in a locked state, and unlocked on transfer. If the secondary chain supports smart-contracts and a sidechain contract is created, then the transfer from bitcoin to secoins will be fully automatic, and a federation has no means to prevent it (even if it holds the custody of transferred funds). This is similar as when a user receives a Bitcoin payment: there is no way the user can prevent this, and therefore it cannot be liable for the origin of the funds. This is the case of Rootstock.
The other procedure where regulation may apply is in transfers from secoins to bitcoins. In case a federation is in custody or in case the miners vote in a drivechain, both groups can be targeted by regulators as money transmitters. In the case of a federation, the legality of the federation will depend on the jurisdiction each federation member is located in. In many jurisdictions transferring cryptocurrency is unregulated. However, if cryptocurrency is treated as money, it seems that federation members should perform KYC on the payments that they authorize to go from the secondary chain to the main chain. In the case that a federation member already hold custody of bitcoins in user accounts (such as the case of an Exchange or online wallet), the payments can be restricted to addresses belonging to users of the Exchange/Wallet, and therefore the standard Exchange/Wallet KYC procedures would apply

normally. If all federation members obey the KYC regulations, other federation members can sign the unlock transaction if and only if one of the members announces it owns the exit address and takes responsibility for the KYC. This case would not be very different from a bank authorizing a normal wire transfer to an account belonging a client of another bank, where the receiving bank has the greater responsibility to perform KYC on his own client, and not the sender bank. An approach that simplifies (or maybe completely eliminates) the need for KYC is to allow the peg for transfers that send bitcoins from an unspent output to an address controlled by the same ECDSA private key that controlled the consumed output, and vice-versa. This is possible in Rootstock, as it also uses ECDSA over the same elliptic curve. Therefore, it is proved that the same party has control of the transferred funds in both blockchains, and no transfer between different entities has occurred. Both systems could be combined: free unrestricted transfers where the source and destination address correspond to the same private key, while the remaining transfers require the authorization by a federation member. This authorization could be either dynamic (a federation member must approve each transfer by means of a message sent to the pegging system) or the pegging system stores a whitelist of source and destination addresses, and the whitelist is maintained by the members of the federation.
There exist the possibility that all participants in a drivechain poll for unlocking bitcoins (federation members and miners in case of RSK hybrid design) were to be considered a single entity and treated as such, but we see this highly improbable since Bitcoin miners are not considered by US regulation as a single entity, even if they collectively enable transactions between different parties.

Conclusion

The best choice for a 2WP design for Bitcoin depends on the properties and features of the secondary blockchain. When the secondary blockchain is merged-mined, sidechains provide no additional security and a **drivechain** seems to be the best choice. Also the merge-miners must run secondary blockchain node and the cost of running an additional voting software module is negligible.

If the secondary blockchain has a federated consensus, but the federation has no interest in the peg, such as when the secondary chain is a private blockchain with smart-contract support, then a **sidechain** is the best choice.

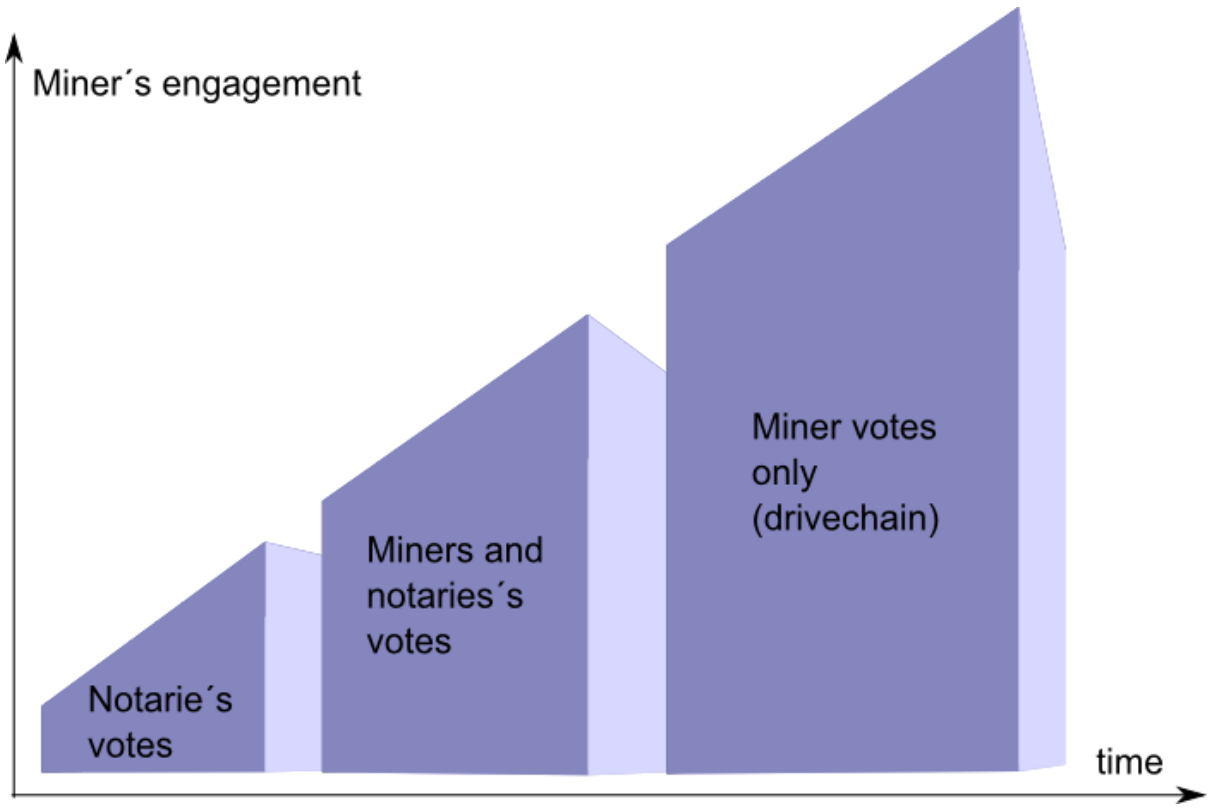
If the secondary blockchain is mined with a different proof-of-work function, then a **sidechain** is the best choice, but it is highly improbable that this case be a leading force pushing the implementation of sidechains in Bitcoin. If the secondary blockchain has its own native speculative token, the Bitcoin community would hardly accept a fork that increases the value of this token. If the secondary blockchain does not have a native speculative token, then it is extremely difficult such a proof-of-work blockchain would be bootstrapped without subsidy.

If the secondary blockchain has a federated consensus, but the secondary blockchain is heavily related to the first blockchain, such in the case of Blockstream's Liquid, then a simple standard **multi-signature custody** is preferred.

In case of Rootstock, where openness and acceptance by the Bitcoin community is crucial, following choices are preferred:

- A sidechain in the Rootstock side
- A hybrid drivechain+federation in the Bitcoin side combining miners and federation votes.

The sidechain proposal in the Bitcoin side has higher probability to be contended by the community or core developers than of a drivechain, as the complexity of the sidechain implementation is significantly higher than of the drivechain. In the case of Rootstock, we must opt for the simplest solution both conceptually and technically in order to maximizes the chances of acceptance. RSK Labs also aim to provide a smooth upgrade path to a fully decentralized solution, so Rootstock plans to start with a federation composed by renowned Bitcoin parties having secoin bonds, and plans to add miner's votes (a drivechain) when available. Using federation votes in the drivechain the system can automatically decrease the number of votes required by the federation and increase the number of votes required by the miners as the amount of merge-miner engagement increases in relation to the amount of bitcoin mining, until no federation votes are required. Once the merge-mining gets almost full miner acceptance (such as 95%), the federation role in voting can be disabled, and only the drivechain remains. The following diagram shows which votes would be required in an hybrid model to unlock the btc at different stages of merge-mining engagement:



Requiring federation bonds and rate-limiting the unlocking of btc adds more security, but also adds more complexity, so these additional security methods are not currently adopted by Rootstock. If Bitcoin soft-forks to provide sidechain functionality, then Rootstock can easily adopt this standard using virtual Bitcoin-like Merkle trees. Therefore RSK Labs will help push both drivechain and sidechain BIPs through the Bit2 project, and RSK will adapt the Rootstock blockchain to the technology the Bitcoin community finally accepts.

Special thanks to Paul Sztorc and Gavin Andresen for reviewing this paper and sending useful comments.

References

[1] https://blockstream.com/sidechains.pdf
[2] https://bitcointalk.org/index.php?topic=91843.0
[3] http://www.truthcoin.info/blog/drivechain/