RavenQt SIG Meeting Minutes for 2021-05-12:

1) Discussion of known problems
    - P2SH patch, Sweep, other

2) Discussion of overall bounty issues

3) Discussion of promoting testing

4) Current thoughts on release plans

===

- Today’s meeting was shorter than usual. We agreed that the top priority was additional testing of v4.7test1

- We briefly discussed the problem in which v4.7test1 cannot sync on mainnet
    - mainnet contains some non-standard transactions with outputs which look similar to P2SH but don’t work properly
    - eg: txid=39aaaed0ccd038e9304e230ecba7b2b9457252cc0b4deb7245d4c2e03a186564&decode=1
    - output:

      ```
      "asm": "OP_HASH160 52fe340f785ad6170d855a2d07ec54e6b7b44861 OP_EQUAL 0 0 OP_RVN_ASSET 1572766e74085343414d434f494e00e1f5050000000075",
      "hex": "a91452fe340f785ad6170d855a2d07ec54e6b7b44861870000c01572766e74085343414d434f494e00e1f5050000000075",
      -"transfer_asset", "name": "SCAMCOIN", "amount": 1.00000000, "addresses": "RYTJ5Tp6yn9GZgo8Q9UXGtqxXBSnBfwuPF"
      ```

    - The code required OP_RVN_ASSET to be at offset 25 in order to recognize the asset script. It also requires the entire ScriptPubKey to be <31 length to recognize a P2SH. The Op_0 padding makes the asset script visible so that the asset gets transferred. But due to the length, the code still misinterprets it as a P2PKH output. Because of that it takes the P2SH redeemscript, ignores the first byte, appends the opcode 87 as the last byte, and misinterprets that as a HASH160 from which it calculates a P2PKH address. So the asset gets transferred to an address which nobody has the keys to.
    - The code in the P2SH PR freezes sync when it sees this transaction

- We discussed the Sweep PR.
    - It is not useful for most users since it requires a full set of indexes on the raven-qt node
    - It can be useful for some service-providers who run a full node, so we can choose to leave the PR in
    - However, the implementation has some known bugs found, such as restricted asset support
    - There have been suggestion to re-implement using an external API, but such things are generally discouraged on nodes for privacy reasons
    - There have also been suggestions of re-implementing using a key import followed by a rescan. That would be a much better way, but is a complete re-write.

We briefly discussed the request for Unique asset support at root

-This would require a chain fork

-We debated the value and difficulty, but arrived at no conclusion

We discussed the fact that Ravencoin has historically implemented all chain forks as hard forks while bitcoin mostly does soft forks

-We agreed that although this may not be ideal, making major changes to consensus code at this time is not worth the risk. A refactoring may be appropriate after Ravencoin is much larger and has more devs. For now, the code works and focus should be on features and applications

-We discussed the fact that soft forking on Ravencoin is made even more difficult if it is desired that the new functionality look similar to the same functionality on bitcoin. For instance, it may have been possible to implement P2SH on Ravencoin as a soft fork by finding a Pay-to-Anyone transaction format in the old code implementation, and then modifying that to support P2SH. But the resulting Ravencoin new P2SH transaction format would look nothing like a bitcoin P2SH transaction

-We agreed that staying as close as possible to the bitcoin codebase is a worthwhile goal so that Ravencoin can try to benefit from bitcoin dev work whenever possible.