API:

* Arguments:
  + Phone number list
* Response:
  + BCH address to send funds to (generated via xpriv protocol)

For every request, bch address and the address list are stored in an sqlite database as a ro

Every 10 minutes:

* Get latest block height
* Scan all txs in blocks range(lastCheckedBlockHeight-6\_\_\_\_\_LatestBlockHeight)
  + Check if any of them send to bch addresses in db
  + For each that does:
    - Check if utxo is unspent
    - Generate invoices for each phone number associated with that address using the utxo amount (minus fee) divided by the amount of phone numbers
    - Pay each invoice using funds
    - Remove row from database
  + For every block in the list, updatelastCheckedBlockHeight to the height of that block
* Every 100 scans, check if fee address has high enough balance to justify payout to me and AW, if so, create payout tx to us with half the funds to each of us
* Exposed API
  + Limit on amount of phone numbers/tx to 2k (based on maximum tx size limit, very conservative)
  + Minimum one phone number per request
  + Rate limit on the tornado server per ip address
  + Validate phone numbers before adding to db
  + Implement https for api
* Error logging
  + ~~Log all failed transactions including address and phone numbers and other errors to error file~~
* Rescan
  + ~~Add table for re-retrying failed blocks, txs and outputs until succeeded~~
  + ~~Do not delete address and phone numbers if transaction fails, wait for it to be re-scanned afterwards~~
  + ~~Even if block fails, update lastcheckedblockheight, since block is going to be re-scanned later~~
    - ~~If tx output or transaction fails, do not exit block, just record it in re-try table and keep going~~
  + ~~Limit on the amount of times a scan can be retried.~~
  + Implement rescanning function that runs once a loop
* Refund
  + Check for invalid phone number error from cointext api, if so, remove row and refund transaction if funds cover fees
  + Check fee rate on tx before broadcast, ensure >= 1 sat/byte
    - Refund if so
  + All refund txs need to have enough funds to pay miners tx fee
    - Ensure sending amount is small enough to accommodate for txfee
* Consumed API
  + ~~Use different api to check if utxo is spent, current one returns true if there are any other errors.~~
* Transactions
  + ~~Check fee rate on tx before broadcast, ensure >= 1 sat/byte~~
    - ~~If it isn't, check if there is enough to refund it, and do so if there is~~
      * ~~This shouldn’t happen because we should calc the fee amount to ensure it is above this threshold, but if it does happen ^~~
  + Maximum age of record
* Fee distribution:
  + Every 100 scans, check if fee address has high enough balance to justify payout to me and AW, if so, create payout tx to us with half the funds to each of us
* Etcetera
  + Centralise all settings into persistence table