Rising Tide

Angular.js

* Rising Tide is a project that will create an asset using the colored coin protocol, and exchange that asset for a set amount of bitcoin each. This will all be done using the blockchain, so the website/web app needs to be capable of monitoring the transactions and delivering a good/service when payment is received.
  + Learn how to create a test asset using the colored coin / Open Assets protocol. For testing purposes, create 1000 units of a test asset RT-test
  + Learn how to monitor transaction of that asset using a bitcoin block explorer
    - Note: Most bitcoin block explorers do not reveal the transmission of colored coin assets. CoinPrism’s blockexplorer does though. <https://www.coinprism.info/>
      * Example Asset <https://www.coinprism.info/asset/AQKbkUkQjKLe7ru44MhgkU8UnnrDUSEqjn>
    - API for monitoring Colored Coin transactions using CoinPrism’s bitcoin block explorer <http://docs.coinprism.apiary.io/#>
      * This will be used in our Web App to see if asset has been transmitted correctly
    - Look into “Colu” , a different colored coin implementation of the Open Assets protocol. It stores data completely in the OP\_RETURN field of a bitcoin transaction. Other colored coin implementations only stored partial information in the underlying bitcoin transaction. I would like to use an implementation that uses the OP\_RETURN field.
* When a user plans to exchange bitcoin for the rising tide asset token, there needs to be the possibility of generating a private key client side in the browser.
  + The “public key” hash of the private key should be the address that the rising tide asset gets sent to, after our system has detected that they sent bitcoin to the proper address.
    - When user sends bitcoin we need to monitor that the transaction has been broadcast. Find a block explorer with API that works for this, consider <https://chain.com/> ‘s API.
      * Rising Tide tells the user what address to send to, this address may have been generated on the spot at that point in time (server side), for that user and only that user’s session.
      * The browser can here be constantly monitoring a payment to that address via the chain api, of which only the user knows about.
      * When the proper amount is broadcast to that address, Rising Tide needs to send some of the colored coin to a new address it creates and lets the user download.
        + Private keys created in browser client side and put in a downloadable format for the user (the user can then add this private key to their own wallet software), the public key address from that private key generation is sent the Rising Tide server, which then sends some colored coin to that address.
      * Rising Tide server knows how much colored coin remain, and how much bitcoin it has received. (It needs to count the bitcoin amounts in all the addresses it has server side.)
      * It needs to update the information on the website about the status of the crowdsale. (X colored coin units remaining, Y bitcoin acquired)

Server Side

* Use AWS EC2 or Digital Ocean or different provider to set up a server. Use state trading society account
  + [operations@stsltd.co](mailto:operations@stsltd.co) ek7hay6t+
* Investigate how much RAM is necessary to generate lots addresses and make RESTful API requests. I think 2-4gb is good, but I am not sure.
* Use node on this server.... or a different framework... research necessary. Should be able to communicate to the client side website via REST