FiatDex Protocol

A decentralized method to trustlessly exchange Fiat currencies for cryptocurrency using non-custodial swaps and game theory.

Background: Currently, there are limited trustless method to exchange fiat currencies such as USD, EUR and others for cryptocurrencies such as Bitcoin, Ethereum or Litecoin. Most users seeking to convert fiat into crypto must use and trust a centralized gateway that serves as gatekeeper for entering and exiting crypto from fiat. Such gatekeepers may require information that users may not want to disclose such as Social Security Numbers, Driver's Licenses or Passports for fear that the information may not be held securely or will be abused. In addition, these centralized gateways can become honey pots waiting to be hacked. As seen with LocalBitcoins, there is a demand for a service that avoids the gatekeepers; however, as we have also seen with LocalBitcoins, such centralized services may become victims to burdensome regulations, whether it is right or not. There are decentralized exchanges that have fiat as an option, such as Bisq or LocalEthereum; however these types of exchanges use shared custody multisignature addresses with a trusted third party that must be the final arbitrator of all trades if the trading parties cannot resolve disputes themselves. Also, if the arbitrators disappear from the platform, the exchange can no longer function when there is a dispute.

What is FiatDex and how is it different? FiatDex is Ethereum based fiat to Ethereum (ETH) swap smart contract. It can be ran as a backend protocol for any Dapp that is order matching users who want to swap fiat for crypto or vice-versa. It utilizes some features from various types of decentralized exchanges and puts it into a simple interactive smart contract. Ethereum was chosen versus Bitcoin as it has more flexible smart contracts and eventually users can swap their Ethereum for Bitcoin via other trustless methods if desired. FiatDex will be the decentralized gateway for users with fiat to enter the cryptoworld without having to trust a third party. The core tenet of FiatDex is **incentive**. Participants are incentivized to continue to the next step, and failure to do so in a timely manner will result in a penalty. Much like Bitcoin and cryptocurrency in general, incentive drives miners to mine and stakers to stake. Incentive will also drive FiatDex. Traders are not asked to implicitly trust each other but rather act out of their **own self-interest**.

How does FiatDex work: FiatDex is a contract with simple steps; however, there is a caveat regarding its use. Both traders must already have ETH to initiate the swap. Now let's start here:

Meet Alice and meet Bob. Alice has 1 ETH she wants to sell for $200 USD. Lucky for Alice, Bob has $200 USD that he wants to use to buy 1 ETH.

1) Prior to the trade, Alice and Bob communicated that they want to exchange and how to do the exchange, including sharing a common hash between each other that will be used as a trade ID. This communication can be Telegram, IRC, email or anything else.

2) Alice creates a swap position using the trade ID in the smart contract with a value of 1 ETH, this swap position requires a 150% collateral, so Alice must send 2.5 ETH in total.

3) Bob sees that Alice has opened the swap position with the correct amount and then sends his collateral to the same swap position which is the same at 150% of the sent amount, so Bob must send 1.5 ETH as well.

4) The swap position now contains 4 ETH.

5) After sending his collateral, Bob must now send $200 USD to Alice. The preferred method is some sort of irreversible cash transfer (not bank to bank as those tend to be reversible).

6) When Alice receives the money and verifies it is not fake, she will then close the swap position unilaterally. This automatically triggers payment from the smart contract to Bob for 1 ETH as well as returning the collaterals to each respective party. Alice will get her 1.5 ETH collateral back and Bob will get his 1.5 ETH back. Both parties are happy and Bob is 1 ETH richer (though $200 poorer) than when he started. Alice is $200 richer (though 1 ETH poorer) as well.

7) If it takes too long to close the swap position, the contract will deduct a fee per time unit that goes to the contract owner address. This is done to encourage the parties to do the swap as fast as possible.

Sounds simple, but what happens if...

- Alice doesn't open the swap position
    - Then Bob doesn't send his part of the collateral as he can see that there is nothing in the swap position on the blockchain. No one loses ETH or fiat.
- Alice opens the swap position and Bob hasn't any ETH into it
    - Alice can refund from the swap position at any time as long as Bob has not put his collateral into the swap position. For refunds, no fee is charged to Alice no matter how long she waits.

- Bob doesn't send the cash or cash is not real / not enough
    - Alice will not close the swap position until she is satisfied. If she never closes the swap position, Alice will have lost an expected 2.5 ETH and equivalent (1.5 ETH from collateral and $200 USD as ETH equivalent). Bob will have lost an expected 2.5 ETH (1.5 ETH from collateral, 1 ETH from Alice). It is in both their interests to finish the manual transfer.
- Alice receives all the money, but declines to close the swap position
    - Alice will have lost an expected 1.5 ETH and equivalent (gained 1 ETH in cash equivalent, gave up 1 ETH to Bob and lost 1.5 ETH in collateral). Bob will have lost an expected 2.5 ETH (1.5 ETH from collateral and 1 ETH from Alice).
- Alice drags her feet when closing the swap position or Bob takes too long to send the cash
    - The contract will automatically start to charge a fee percentage after a certain period of time has elapsed since open, from the collateral amounts of both parties equally. This fee is deducted when Alice closes the swap position. It will never touch the sending amount. The fee goes to the contract owner. The fee is there to incentivize the traders to complete the swap as quickly as possible. In version 1, the fee starts to apply after 7 days and increases 1% per day thereafter. The largest the fee can ever be is 99% the value of the collateral.

FiatDex Gateway: FiatDex Gateway is a simple client based web interface with the FiatDex protocol. It requires MetaMask to use and is a useful starting point if you plan to integrate FiatDex protocol into your own Dapp. The website is best ran on a localhost server as MetaMask requires a server connection.

FiatDex Protocol is usable now

Visit etherscan to understand the contract functions and to implement the contract into your own Dapps.

ETH Mainnet Address: 0x2c110867ca90e43d372c1c2e92990b00ea32818b

ETH Rinkeby Address: 0x2c110867ca90e43d372c1c2e92990b00ea32818b

External Contract Functions

openSwap(bytes32 _tradeID, address _fiatTrader)

This function opens the initial swap position. Alice is the ethTrader who calls this function to create the swap. The tradeID is a variable that is shared between ethTrader (Alice) and fiatTrader (Bob). The ETH that is sent to this function is unevenly split into amount being sent to fiatTrader and amount being held as ethTrader collateral. Also, the clock starts now for the swap to close if traders do not want to incur a fee.

addFiatTraderCollateral(bytes32 _tradeID)

The fiatTrader sends ETH to this function to add collateral to the swap. Only the fiatTrader can add ETH into the swap position. The ETH required must be equal to or greater than the collateral already present from ethTrader otherwise the transaction will fail. Once the fiatTrader has done this and confirmed a successful transaction, the fiatTrader needs to now send fiat to the ethTrader.

refundSwap(bytes32 _tradeID)

This function will refund the ETH from the ethTrader if the fiatTrader hasn't already put in his collateral. The ethTrader cannot refund if the fiatTrader has already put in collateral into the swap position. No fee is charged regardless of the time.

closeSwap(bytes32 _tradeID)

This function will complete the swap position. The ethTrader cannot complete a swap position unless the fiatTrader has put funds into the collateral. Only the ethTrader can close the swap. The collateral is returned to each trader and the fiatTrader will get the sent amount from the ethTrader. The ethTrader should only close the swap if she has **confirmed and verified** the fiat she has received from the fiatTrader. This is also when the fee will be subtracted evenly from the collateral of both traders if a fee is warranted.

changeContractOwner(address _newOwner)

This function is called only by the current contract owner and it will change ownership to the new owner. The only role of the contract owner is to collect fees from delayed closed swaps. The owner cannot destroy the contract or stop the swaps. The first contract owner is created when the contract is created and it will be the address that deployed the contract.

External Viewable Contract Functions

viewSwap(bytes32 _tradeID) returns:

(uint256 swapState, uint256 sendAmount, address ethTrader, address fiatTrader, uint256 openTime, uint256 ethTraderCollateral, uint256 fiatTraderCollateral, uint256 feeAmount)

This function is an ETH Call function that provides information about the state of the swap position to both the ethTrader and the fiatTrader. They will use this information to determine where they are in the trade and what they should do. Possible swaps states are: NOTOPEN (0), INITIALIZED (1), ACTIVE (2), CLOSED (3). NOTOPEN is the default state prior to when the ethTrader opens the swap position. INITIALIZED is the state when the ethTrader opens the swap position, waiting for the fiatTrader to put in the collateral. ACTIVE is the state when the fiatTrader has added the correct collateral. At this point the fiatTrader needs to send the fiat. CLOSED is the state after the ethTrader has closed the swap position regardless of whether the fiatTrader sent collateral or not.

viewFiatDexSpecs() returns:

(uint256 version, address owner, uint256 feeDelay, uint256 dailyFeeIncrease)

This function is an ETH Call function that provides information about the contract in general and the fee specification. version is the version of the contract. feeDelay is the amount of time in **days** the contract will wait before it starts charging a fee. dailyFeeIncrease is the **percentage** increase in the fee **per day (24 hour period)** that will be subtracted from the collateral amounts of each trader. The intial fee percentage is 0% and starts to increase after the amount of days in feeDelay by the dailyFeeIncrease per day. The fee percentage cannot be greater than 99% of the collateral. As dailyFeeIncrease is an uint256

with 3 decimal places, a value of 100,000 represents 100.000%, a value of 1 represents 000.001% and a value of 1,000 represents 1.000%. All day units are converted to seconds and compared to the openTime to calculate the fee.

Date: Thursday, August 06, 2019