

# Pay With Crypto Demo NFT Version 1



The idea with this project is to demonstrate a reasonable solution with regards to allowing someone to mint an NFT using different crypto currencies. A couple different payment options are allowed, but the NFT is minted on the Theta blockchain after receiving payment from a different blockchain.

This software demo is effectively a ‘roll your own’ solution that allows receiving payment on the Ethereum compatible blockchain of their choice yet minting the receipt for the transaction on the Theta blockchain.

The solution is also built upon the Selene Network protocols which ensures that the agents involved in the sale of the NFT receive their commissions.

## Get you one!

As with all my demo projects, this one comes with its own collectable NFT that lets you experience the project for next to nothing!

The mint price is set at \$1. Note that my Selene Network NFT projects all use the Penny Oracle to offer the latest accurate pricing in Tfuel. Yet this project allows for minting in a few different currencies. Each mint comes with it’s own NFT image and metadata.

If you are an agent in the Selene Network, this NFT mint pays 25% in commissions.

## Minting

The visitor to the website is presented with a Mint button that contains the logo of the currency that will be requested for the mint.



Note that the website visitor is paying in one coin and commissions for minting are paid in tfuel, someone other than the visitor needs to be involved in supplying the tfuel to complete the mint. This effectively creates a situation where someone has to watch for the payment and once received, it mints the project NFT for the visitor.

To solve this, this demo uses a Cross Chain Mint Oracle software (CCM Oracle) that monitors for payments so that it can mint the project NFT into the visitor's wallet. This is done automatically in the background after a short delay.

## Details

In order to provide this functionality, a project creator needs to include & implement the IOrderBookV1 interface in their project NFT smart contract. The functions in this interface allow a dedicated wallet that is setup by the project owner the ability to mint a project NFT into another wallet. Because the project owner is the one that receives the funds, the project owner is responsible for managing the balance of the Oracle's minting wallet. The Oracle's minting wallet is the wallet that pays commissions to agents in the Selene Network.

## User Scenario

The user visits the project creator's website and sees that there is both a 'Mint with Eth' and 'Mint with BNB' button presented for purchasing the NFT. The visitor knows they have enough BNB in their wallet to make the purchase thus they click that button.

The website pop's up the Metamask wallet asking to change networks to the BNB chain. The user accepts and then the website launches Metamask asking for a BNB payment. The user validates that the correct amount is displayed in Metamask and confirms the purchase. The website then launches Metamask to restore the network back to the Theta network.

Because the CCM Oracle has to pick up the purchase event and process it, the website shows the visitor that they have a 'pending' order.

After a short period of time (maybe 10 minutes), the CCM Oracle notices that there is an outstanding order and it processes it. It confirms that the right amount was sent to the project creator's wallet and then it issues a mint for the visitor. The mint is a signed transaction broadcast to the Theta Network.

Later, when the user refreshes the project page, the 'pending' order no longer exists and their wallet holds a project NFT.

## Technologies

The current implementation of the CCM Oracle is that it is a console app on windows that is always running and checking for orders. Because it needs to sign a transaction that will be broadcast to the network, it uses PHP code that runs on a local development server. That local server's PHP code needs to hold a private key, thus that part of the project is kept private.

Fortunately, if others want to use this technology, they could either setup a similar development server or just manage a wallet that is used by my development server. The code is designed for the second solution here. The code that generates the mint button on the website will monitor the balance on the CCM wallet used by that button. If the balance falls too low to cover commissions, the button will not be enabled.

Thus, the cost of accepting payment on one currency and then minting in tfuel is the cost of maintaining the CCM wallet that is used to perform the mint.

## Credits

Would like to give thanks to the folks over at [ThetaScan.io](https://ThetaScan.io). Without the transaction broadcast functionality that that website provides, this CCM Oracle would not be possible.