Andrea Stojanovski
Jordan Stojanovski

91 lines (48 sloc) | 6.81 KB ···

NFT Lending Auction

Introduction

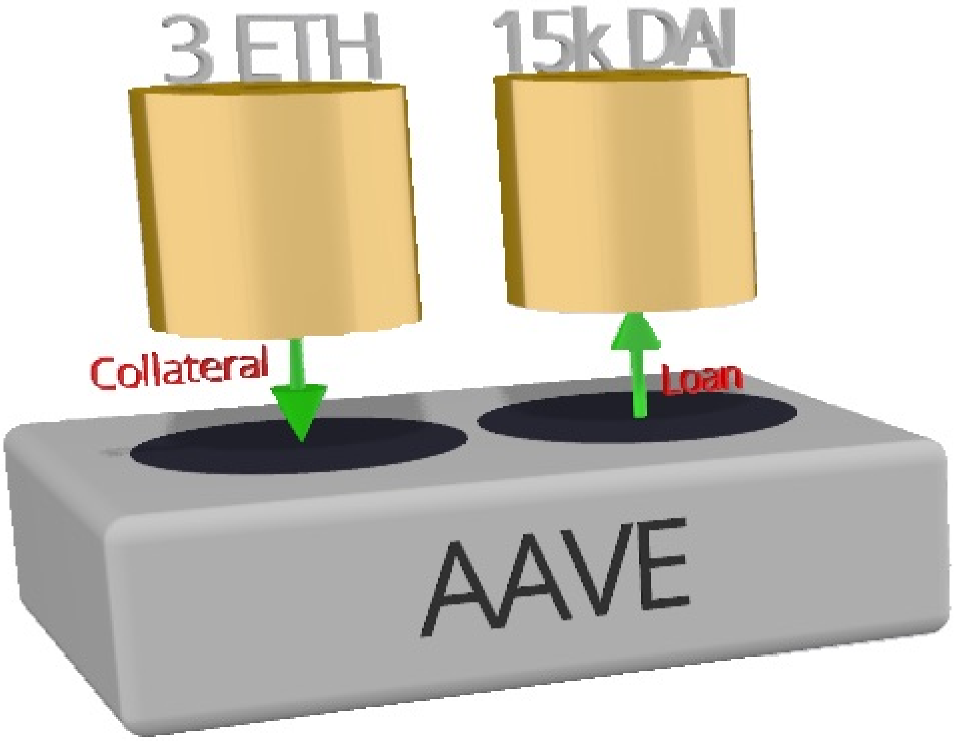
This is a decentralized application for lending Non-Fungible Tokens (NFTs). The web front end is written for deployment on IPFS. As such it can be loaded as static content and it does not rely on any active service. The back end is a Smart Contract running on the ALgorand blockchain.

This application allows both the Borrower and the Lender to be engaged in Decentralized Finance (DeFi) within and outside of this application.

In an auction the prospective Lenders compete for the best offering to the Borrower. The Borroser uses his NFT as collateral for the loan, to receive funds that can further be engaged in other DeFi protocols.

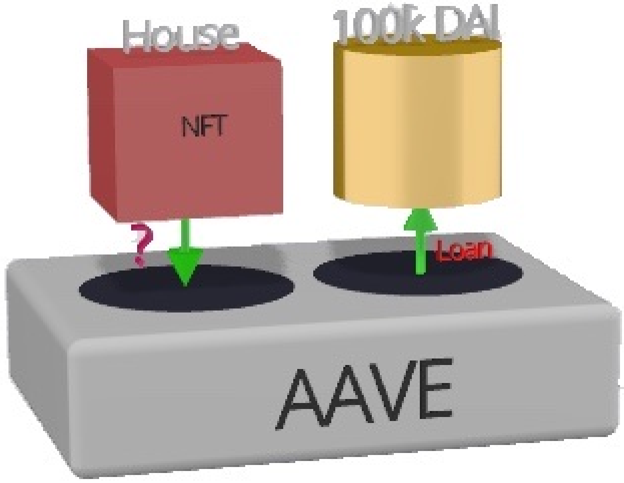
Non-fungible Lending

Using fungible collateral in DeFi lending was one of the first offerings. Typically, the borrower vouches a collateral which is expected to increase in value. Against that collateral, the borrower gets a more stable asset, which is expected to be easier to repay. The funds borrowed can be put to work in other DeFi protocols. For the lender, the profit lays in the interest rate charged for the assets lent. Lenders can chose between various lending protocols in order to optimize their yield. For example, one of the first and largest DeFi lending protocol is AAVE:

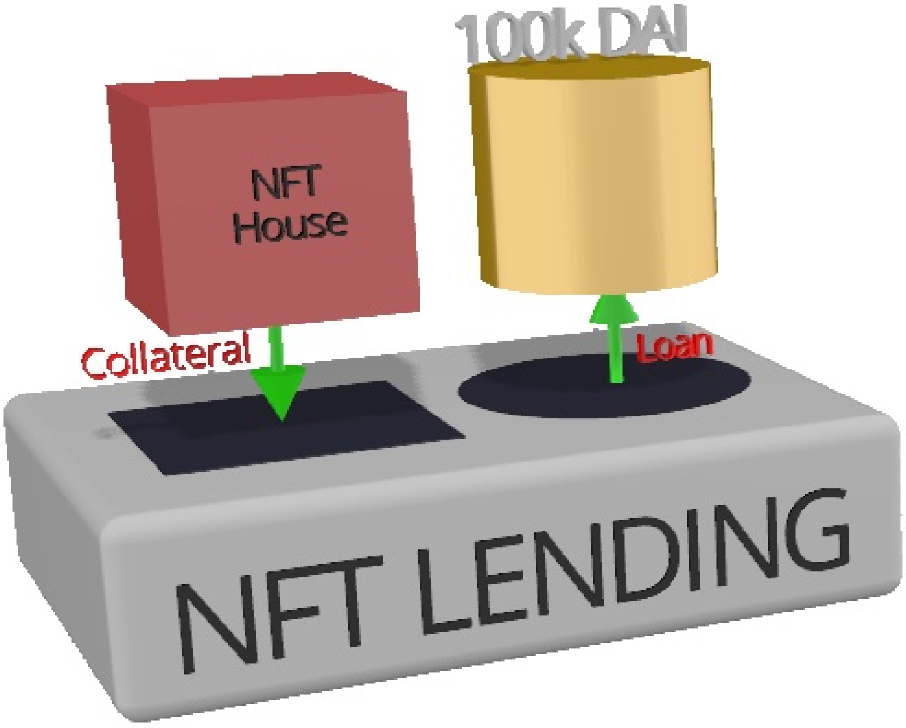


This is very convenient for using fungible assets as collateral. However, if the borrower has liquid fungible assets, the motive for obtaining a loan against that is not very intuitive.

A house for example is non-fungible. It can be represented as NFT. AAVE expects a fungible collateral, as it is easy to treat in "bulk" without special attention put into each loan. A non-fungible collateral would not work there:

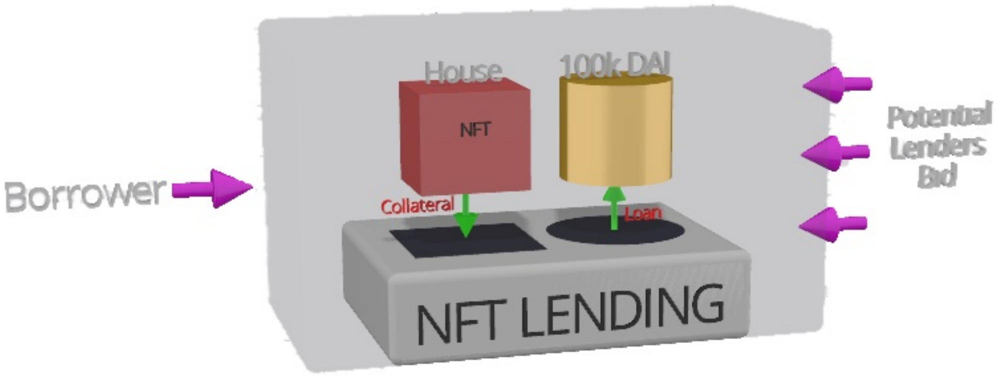


This is why we need a protocol for lending against non-fungible collateral:



The simplest way to do this would be to **find a way the Borrower and the Lender to somehow fin each other** and decide on the terms. Then they can enter a simple escrow Smart Contract to achieve the deal trustlessly. Trustlessly means without a need of trust - all anyone needs to trust is that the Smart Contract is going to behave as programmed, and to assure this is the core job of the Smart Contract capable blockchain such as Algorand.

Instead of sending the Lender and the Borrower to another venue, in order to find each other, a friendly DApp would allow them to stay and find each other without going elsewhere. To achieve this, the DApp offers an auction:



Implementation

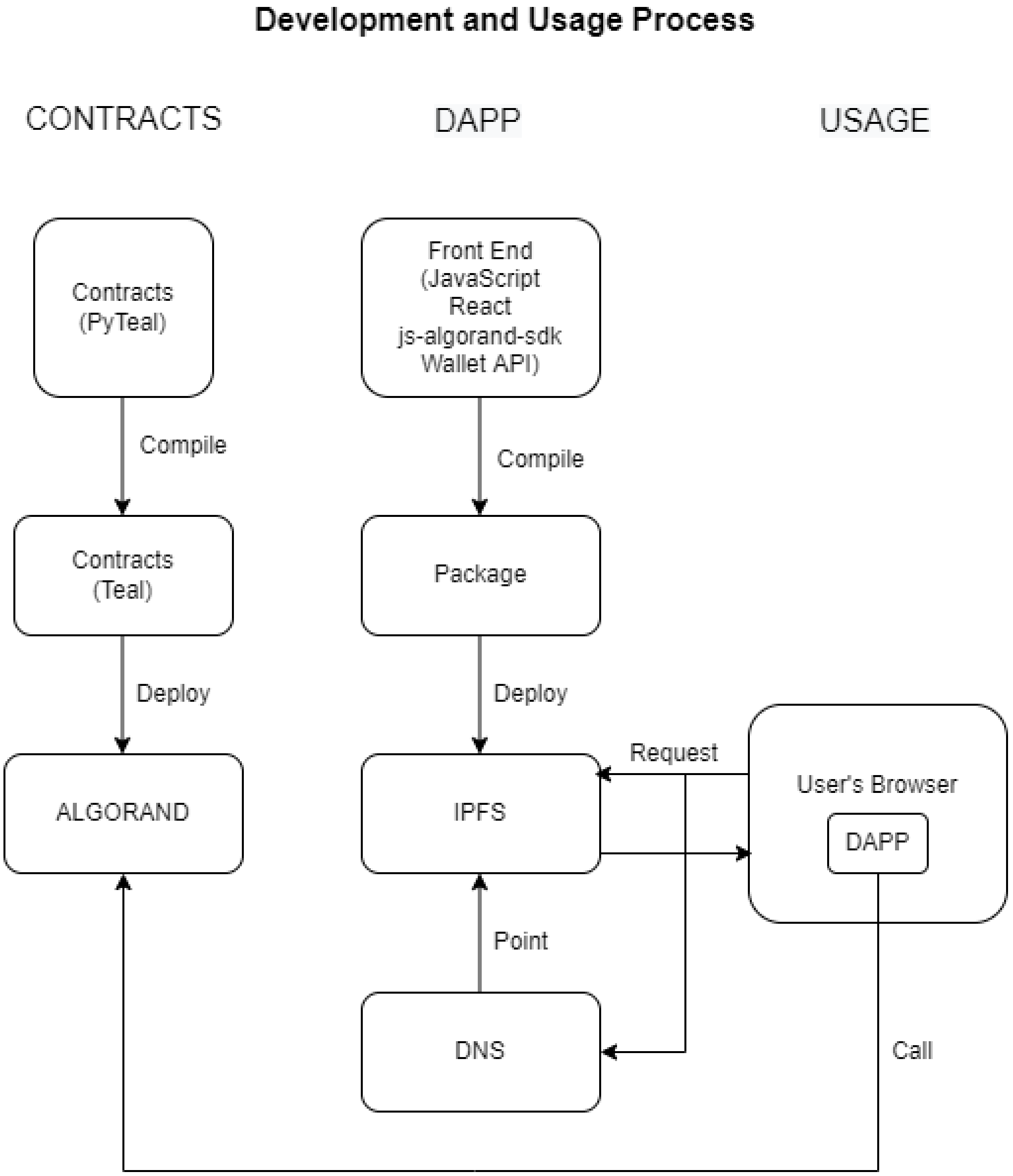
The Decentralized Application (DApp) uses the Algorand blockchain as well as a web interface, accessible over the Internet.

The business logic of the application is implemented as Algorand smart contract.

The front end is running in the user's browser as JavaScript code. This code is served as static content from IPFS (Inter-Planetary File System), and as such it is immutable and not prone to front-end code modification attacks.

Implementation and Usage Lifecycle

The following diagram shows the deployment and usage lifetime of the application.



In the first column, the contracts written in PyTeal are compiled into TEAL code understandable by Algorand an deployed to the blockchain.

In the second column, the front-end, written in JavaScript and using React, js-algorand-sdk and the wallet APIs is packaged and deployed on IPFS. The DNS record is updated to reflect the IPFS URL, so the user can access the application by it's domain name. In the future, in addition to DNS, Ethereum Naming Service (ENS) or Unstopable Domains name sevice shall be used.

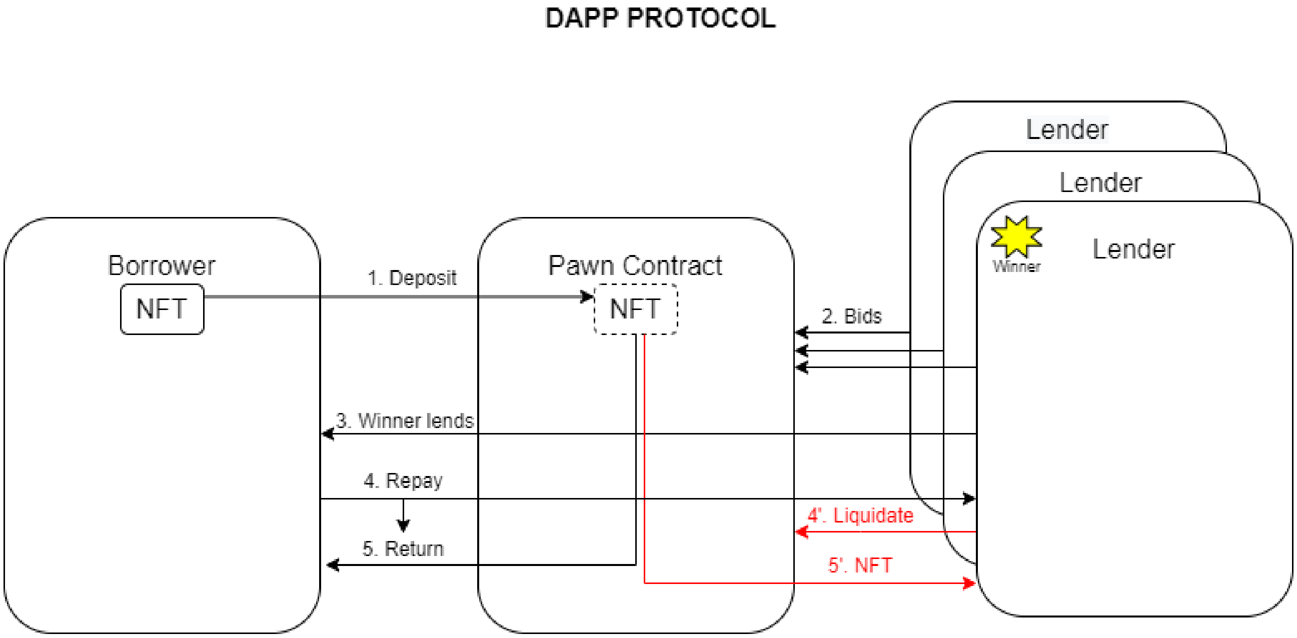
In the third column, the user can access the DApp using a browser. The browser requests the DNS pointed and IPFS hosted JavaScript code as static content. This code then accesses the Algorand blockchain and uses Algorand wallets for signing transacions before sending them to the nearest Algorand node.

Notably, Algorand network cost is relatively low. This allows for the Borrower to deploy a new instance of the Smart Contract for each NFT put as a collateral, and create an auction for it. When the auction ends, the lending commences, and when the lending ends, whether with repayment or liquidation, the Smart Contract can be destroyed.

The auction is governed on-chain as low network fees allow this as well.

Workflow

The NFT lending process is described as follows:

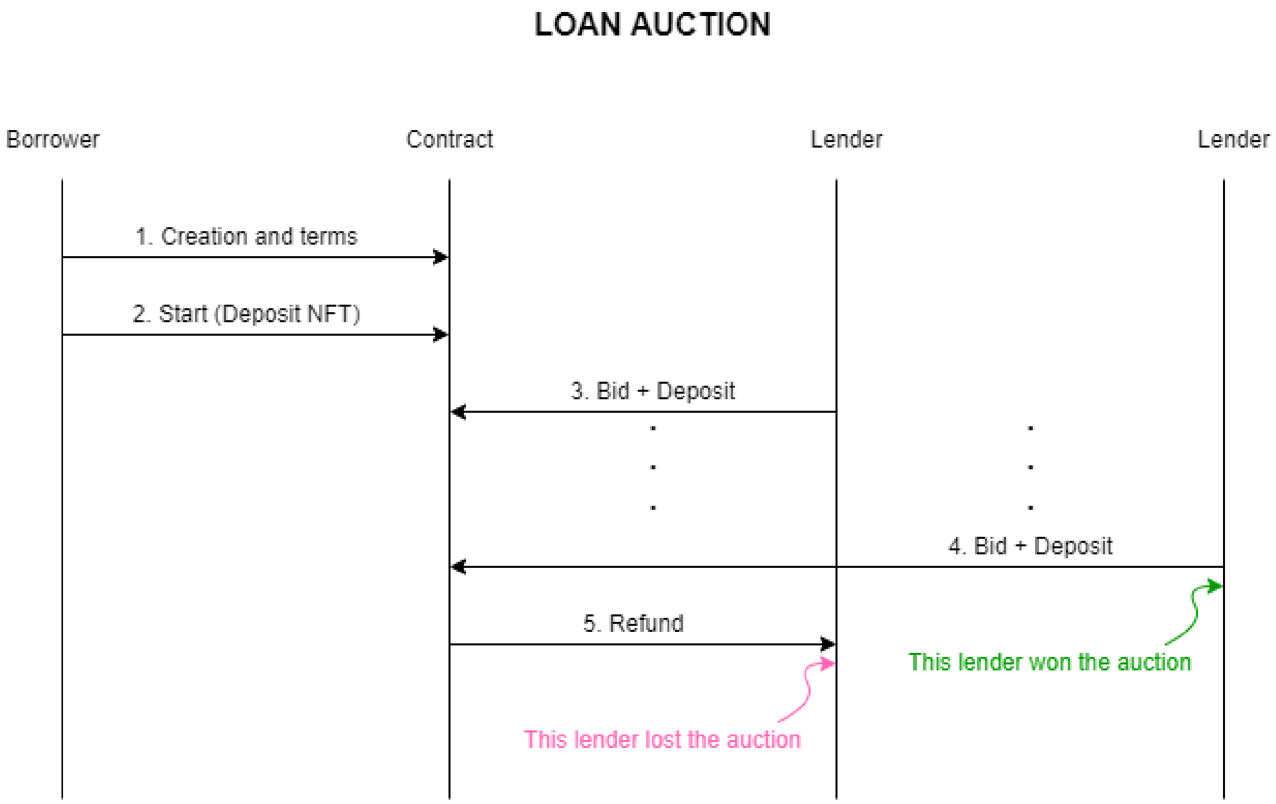


- The Borrower sets up the auction by offering his NFT and putting it into the Auction Smart Contract as collateral. The initial parameters are the duration of the auction, deadline for repayment of the loan, the amount borrowed, the maximal agreeable repayment amount, the minimal bid decrease (factor of the previous difference between the above two parameters).
- Potential Lenders start bidding at the maximal agreeable repayment amount and compete by bidding lower repayment amounts. At the end of the auction the winning bidder is the one offering the lowest repayment amount. As prospective Lenders bid, they deposit the funds to be borrowed into the Smart Contract. As a new winning bid arrives along with a deposit, the overtaken bidder is refunded her deposit immediately.
- Once the auction end, the loan commences. The Borrower can immediately borrow the funds by pulling them from the Smart Contract.
- If the Borrower repays the loan timely before the deadline, he gets his NFT back, while the repayment funds go immediately to the Lender.

- If the Borrower fails to repay the loan timely he goes into default, after which the Lender can liquidate the loan and receive the NFT deposited as collateral.

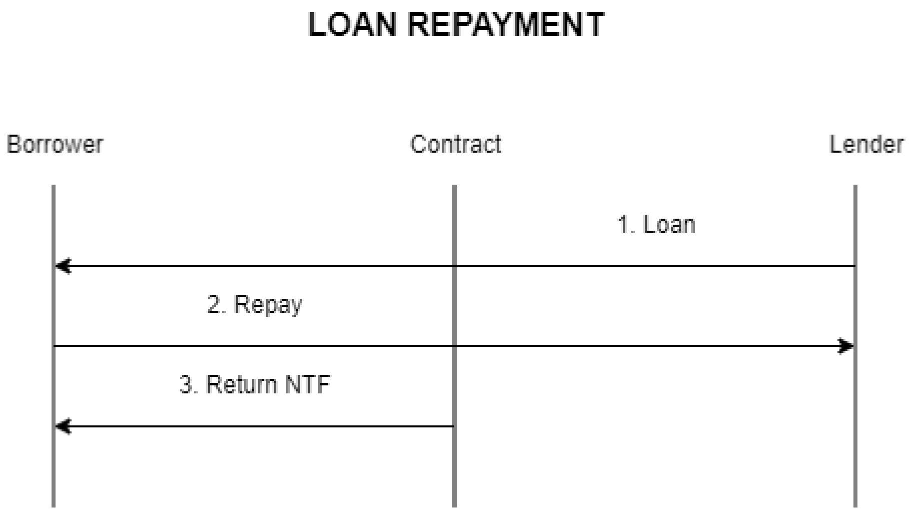
So why does not the Borrower make payments, like in the regular bank loans? There is no need to repay while still having the collateral locked - this is just an additional risk. Instead the Borrower can engage his funds elsewhere in DeFi and make extra money.

Here is the timing of events. Initially there is a loan auction:



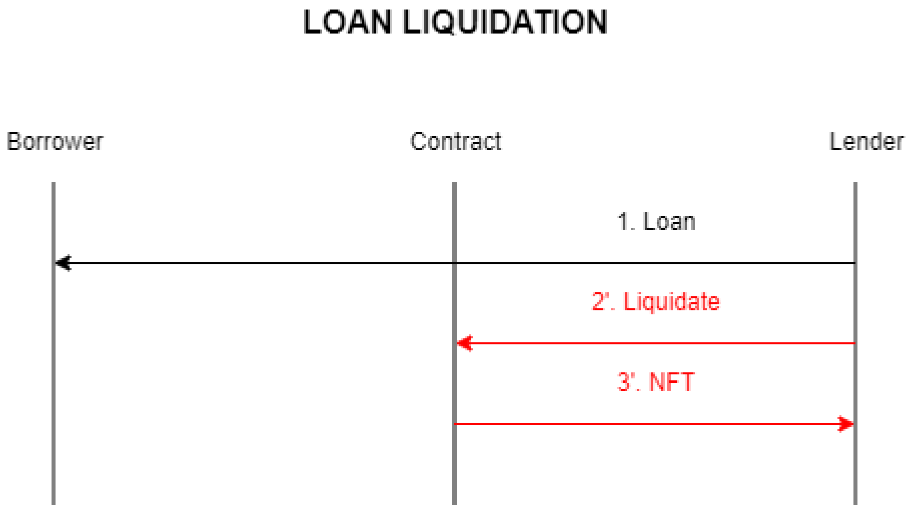
There are two possible outcomes: successful repayment or liquidation.

Repayment:



The owner deposits the NFT along with the terms. The lender accepts the terms and lends the money in form of an Algorand ASA token. Once the owner repays the debt determined in the terms, the NFT is returned to the owner.

Liquidation:



If the NFT does not repay the debt timely within the deadline, upon request of the lender the NFT is transferred to the lender as part of the liquidation process.

Further Work

In addition to ability to set up auctions, an on-chain auction advertising Smart Contract shall be developed. This way Borrowers and Lenders can meet at this DApp, without having been in contact previously in any way.