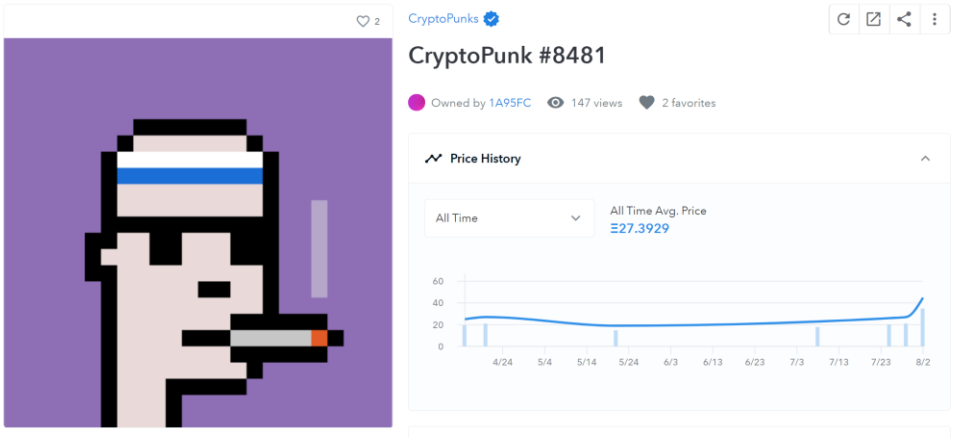
Mktplace

A certificate of deposit auction protocol for limit orders on anything.

Problem

The value of an asset is generally considered to be the price it can be sold for.

For a fungible token the value is easy to identify because there are large liquid markets with ready buyers and sellers doing thousands of transactions a day. But for non-fungible assets, like CryptoPunk #8481, sales are significantly rarer. This punk has only been sold (i.e., priced) 7 times since April 2021.



The value can only be estimated based on past sales of the asset (e.g., the recent 44 ETH sale) and recent sales of related assets (i.e., other sales of the CryptoPunk collection).

There are 2 main problems with this:

1. Markets thrive on information, and ad-hoc pricing of illiquid assets is a messy heuristic and subject to large swings.
2. It makes it difficult for the asset holders to access critical financial services such as *collateralized debt positions (CDP)*.

This paper proposes Mktplace as a **decentralized pawn shop** to solve these 2 problems and unlock significant value for illiquid asset holders who want cash but don't want to commit to selling their asset.

Formal Definition

1. Given a Holder {H} and a Vault of assets {V}
2. Bidders {B} can submit certificates of deposit bid {C} for time {T}.
3. The Holder can use highest live bid { HLB = MAX(C) } as collateral for a loan {L}.

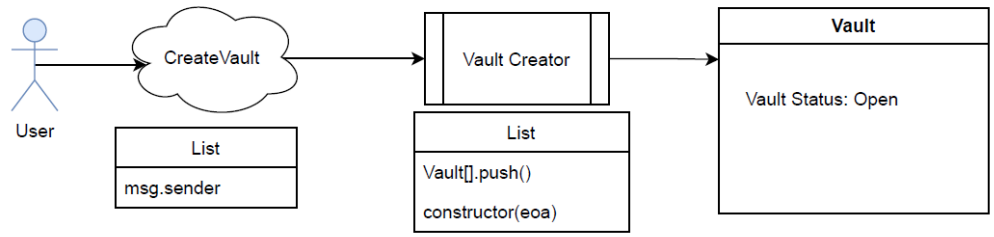
Consider:
Holder {H} has Vault {V} and takes a Loan {L}
Where { $L < C_1$ } AND { $C_1 > C_2 > C_3$ } AND { $T_1 < T_2 < T_3$ }

Bidder {B}	CD / Bid {C}	Time {T}	Status
B_1	C_1	T_1	Open | HLB
B_2	C_2	T_2	Open |Pending HLB
B_3	C_3	T_3	Open

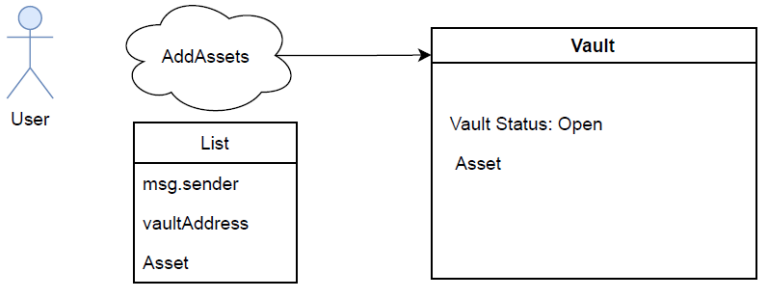
One of 3 events can occur:

1. If the loan ever exceeds the highest live bid, the vault is liquidated and given to the current HLB Bidder. All bids are then closed early with no penalty.
    a. IF ($L > C_1$) THEN { V → B_1, L → 0, B_2 CLOSED, B_3 CLOSED }
2. If the highest live bid expires AND the loan is above the **next** highest live bid, the vault is liquidated and given to the bidder with the **current** highest live bid.
    a. IF (Current Time >= T_1 AND $L > C_2$) THEN { V → B_1, L → 0, B_2 CLOSED, B_3 CLOSED }
3. If the highest live bid expires AND the loan is NOT above the **next** highest live bid, the loan continues with a new highest live bid resulting in a shift of the credit limit curve.
    a. IF (Current Time >= T_1 AND $L < C_2$) THEN { B_1 CLOSED, B_2 → HLB }

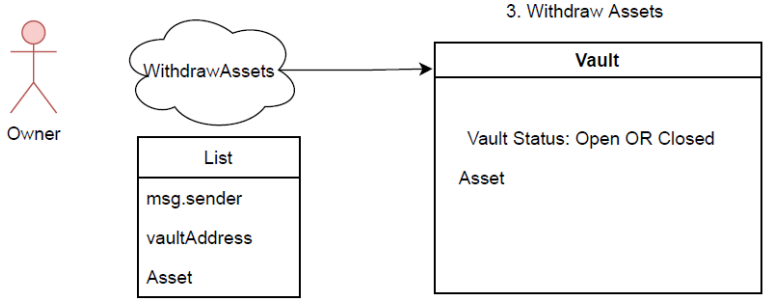
1. Create Vault



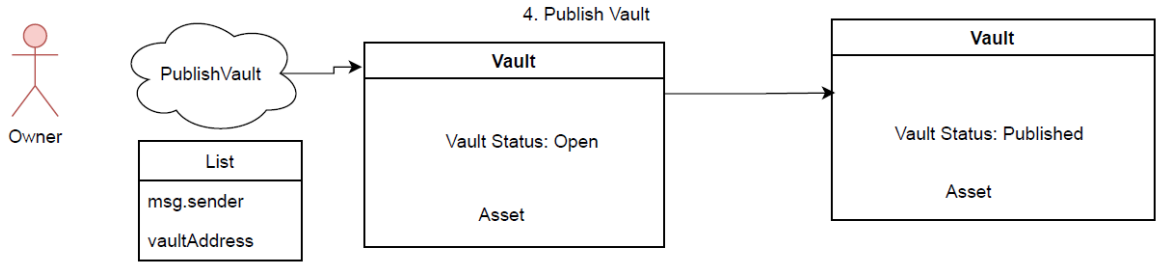
2. Add Assets



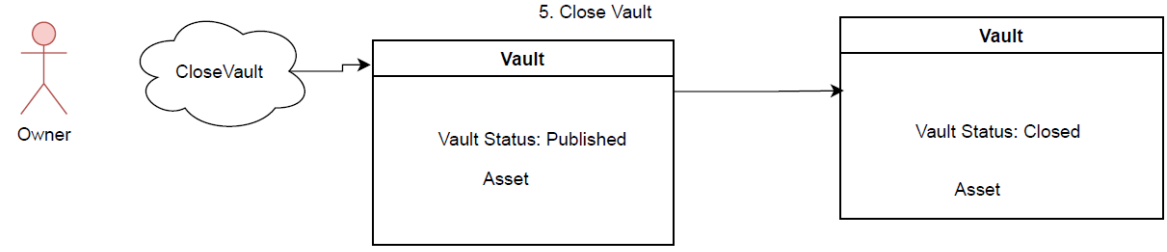
3. Withdraw Assets



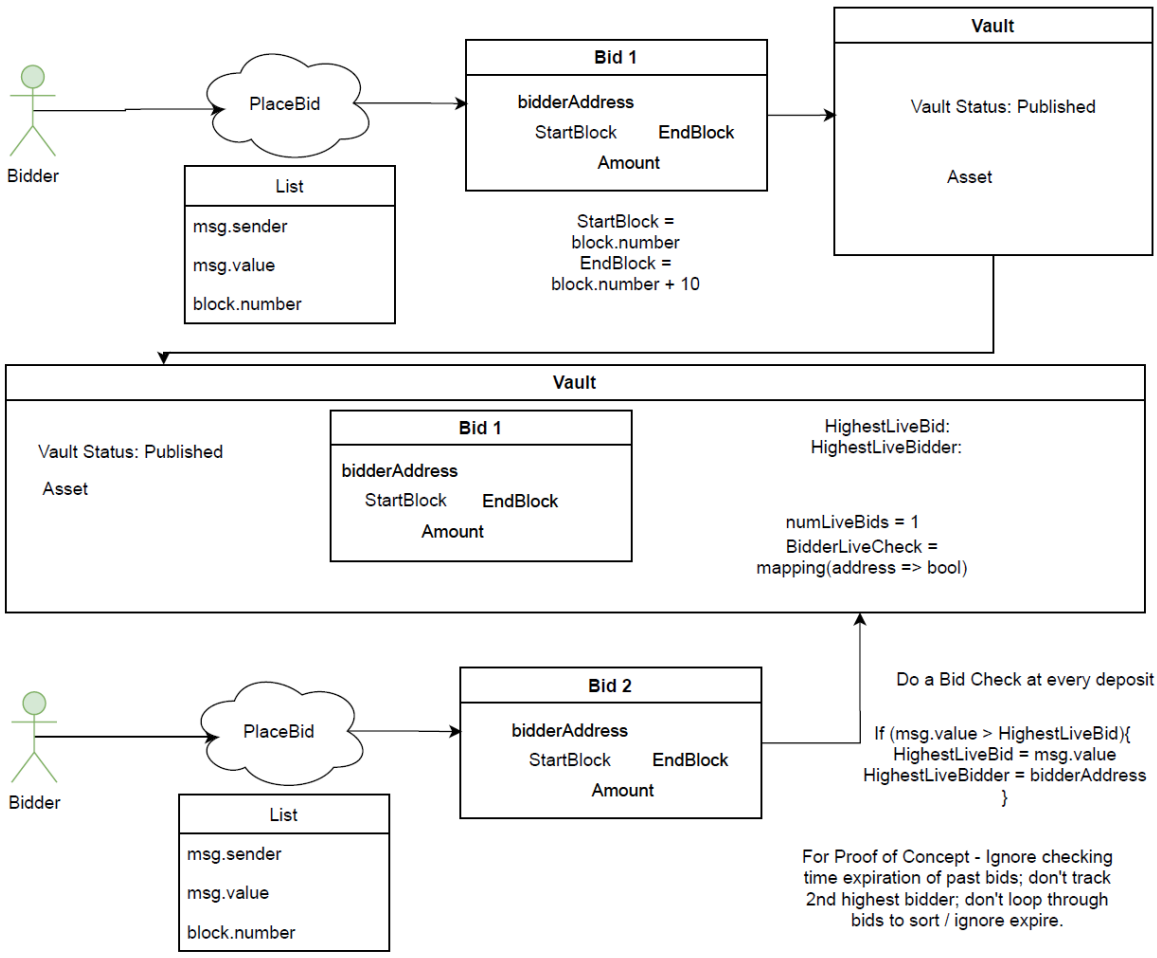
4. Publish Vault



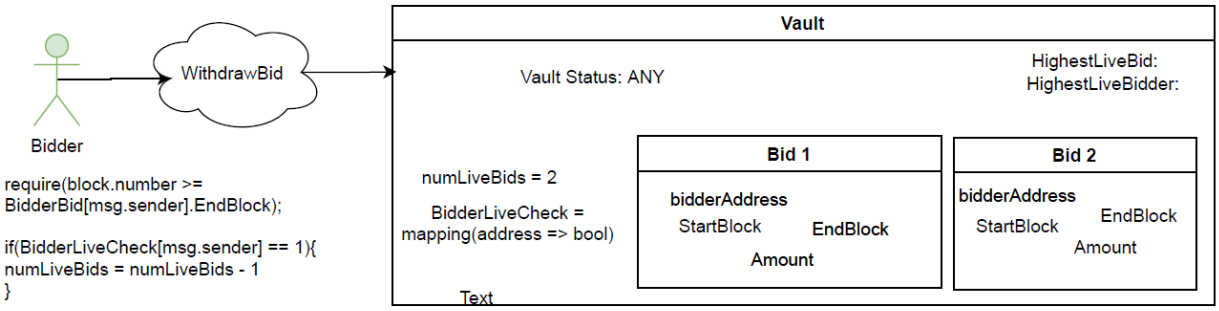
5. Close Vault



6. Bidders Bid



7. Bidder Requests Withdraw & related



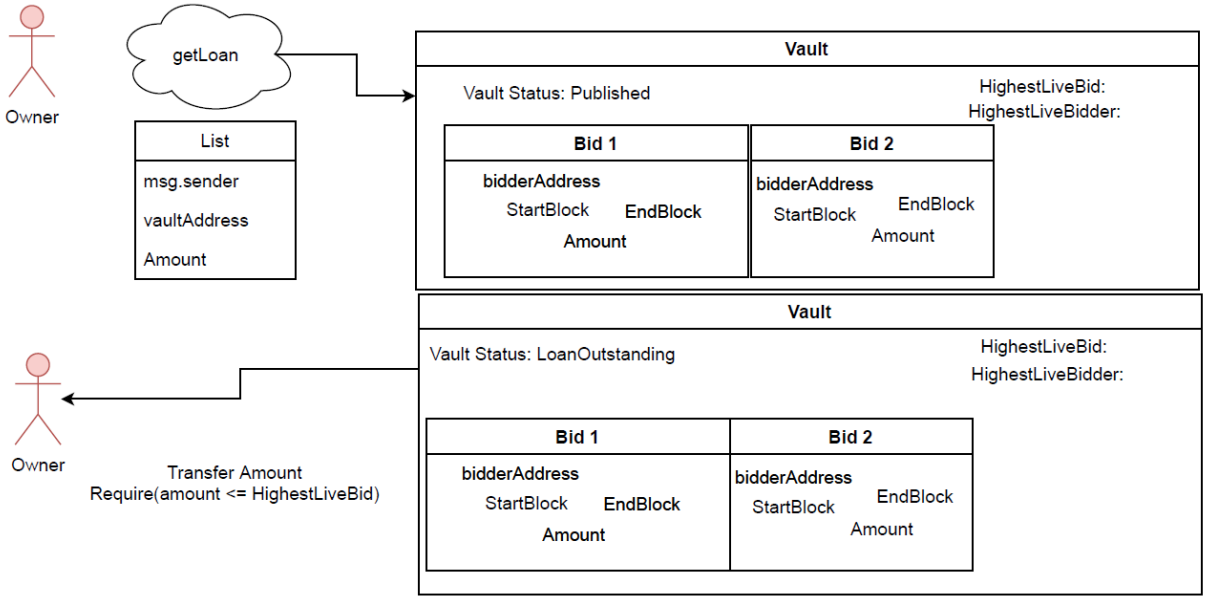
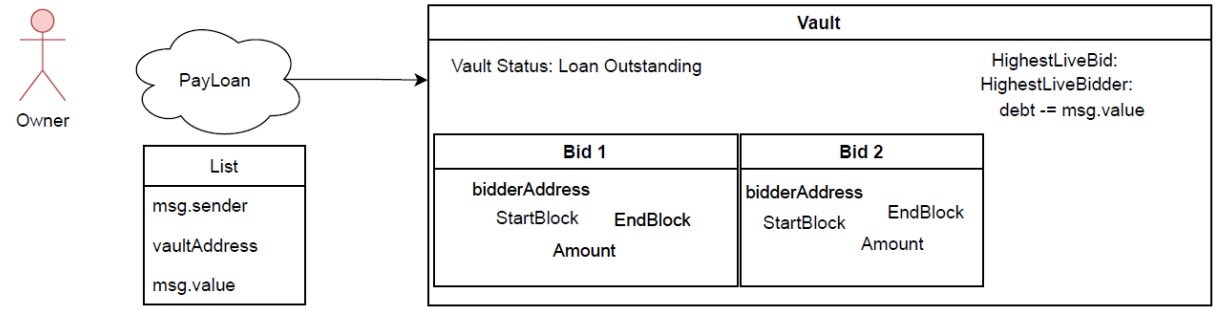

```
require(block.number >=
BidderBid[msg.sender].EndBlock);

if(BidderLiveCheck[msg.sender] == 1){
numLiveBids = numLiveBids - 1
}
```

```
If( vaultStatus == VState.LoanOutstanding){
checkLiquidationEligible()
}
}
```

```
if(vaultStatus = VState.LiquidationEligible){
if (msg.sender == HighestLiveBidder){
BidderLiveCheck(msg.sender) = 0;
BidderBid[msg.sender].amount = 0;
HighestLiveBid = 0
ChangeVaultOwnerToHLB()
vaultStatus = VState.Published
} else Transfer(BidderBid[msg.sender].Amount, msg.sender)
}
```

For Proof of Concept - HighestLiveBid is only updated when highestLiveBidder withdraws; and it only updates to 0. We *won't* pass HLB status for the POC.

8. Owner Takes Loan



9. Owner Pays Loan