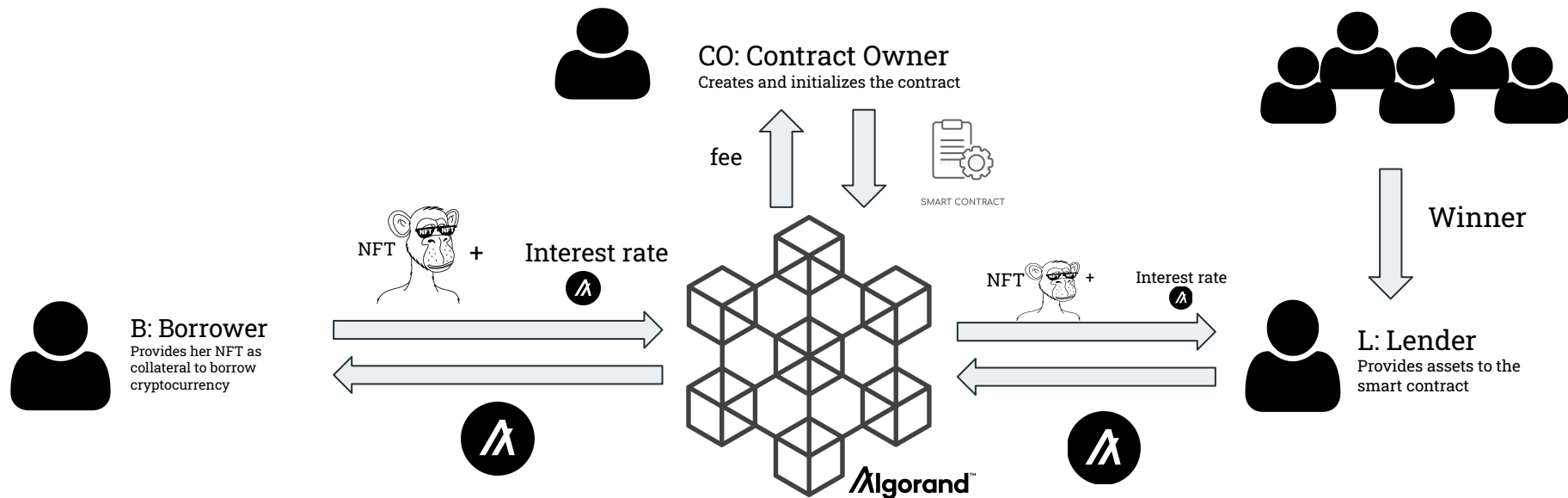Group 3

NFT LOAN

Capocasale Vittorio
Castellano Dario
Renieri Margherita
Toffalini Alessandro

Algorand™ FOUNDATION

Classical pawn shop

Decentralized virtual pawn shop smart contract

CO: Contract Owner
Creates and initializes the contract

fee

SMART CONTRACT

Winner

NFT + Interest rate

B: Borrower
Provides her NFT as
collateral to borrow
cryptocurrency

NFT + Interest rate

Algorand

L: Lender
Provides assets to the
smart contract

State of the Art

Peer-to-peer: individuals borrow and lend money directly from one to another
- Examples: NFTfi, TrustNFT, Pawnfi, Yawww, Arcade

Peer-to-protocol: individuals borrow liquidity from the protocol. Providers offer liquidity to the protocol. The protocol handles liquidations and price esteems
- Examples: NFTuloan, BendDAO, PINE

Smart Contract information 1/2

- **nft_id** (a handle to retrieve the correct NFT)
- **borrower_address** (the address of the borrower)
- **lender_address** (current highest bidder)
- **highest_bid** (highest bid)
- **auction_base** (auction starting amount)
- **auction_period** (auction deadline block number. The client submits a duration, the contract computes the deadline)

Smart Contract information 2/2

- **payback_deadline** (loan deadline. It initially stores the duration. It is calculated after the accept_offer is invoked)
- **last_interest_update_block** (starting block to compute the interest. It stores the block corresponding to the last successful invocation of pay_back)
- **debt_left** (current debt. debt_left=debt_left*(1+interset_rate*(current_block - last_interest_update_block))
- **state** (current contract state: 0=initial, 1=auction started, 2=bid accepted)

Technical challenges

The development of the project led to face and manage several technical challenges:

- Any Algorand Standard Assets can be received as collateral
- Economic concepts (i.e., dynamically calculated interest on the portion of the unpaid debt)
- The debt can be paid back at any moment, in any percentage
- Atomic groups and 0-fee inner transactions
- Minimum balance management with dynamic optins
- The smart contract can be used multiple times in sequence without recreating the app
- We handle multiple scenarios (candle auction, offer cancel, borrower/lender unresponsive, deadlines, etc.)
- Temporal parameters have to be calculated in terms of blockchain round. This led to some issues in Testnet already raised to the algorand repository that has been solved pushing arbitrary round numbers when needed.

Strengths and innovativeness

Many features:
- The contact handles a full working solution including the initial auction, the possibility to cancel the auction (borrower only), the possibility to take back the bid if the auction period expires, the possibility to obtain the NFT if the debt is not paid back, the loan interest and contract fee management
- Candle auction mechanism
- The contract is reusable: it can handle multiple loans in sequence
- Any ASA can be used as collateral
- The debt can be paid back at any moment, in any percentage
- Triple gain: the contract benefits the borrower, the lender, and the contract owner
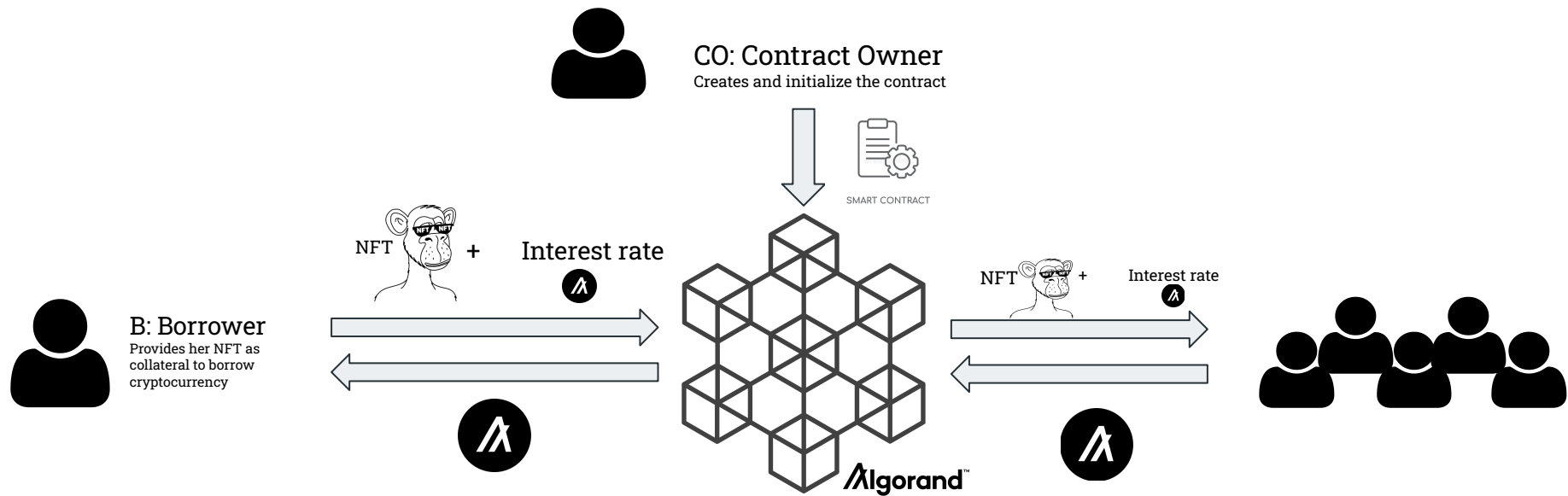- We used Beaker

Limitations

The project is affected by some limitations:
- We can handle a single NFT at a time
- We currently constrain the borrower to offer a single ASA. However, by storing an amount variable in the contract state, we could receive any amount of ASA (e.g., stablecoins)
- We compute a simple interest, not a compound one
- Only one bidder can obtain the NFT (i.e., no shared bids and fractionalized NFTs)

Possibility for the future

Thanks!