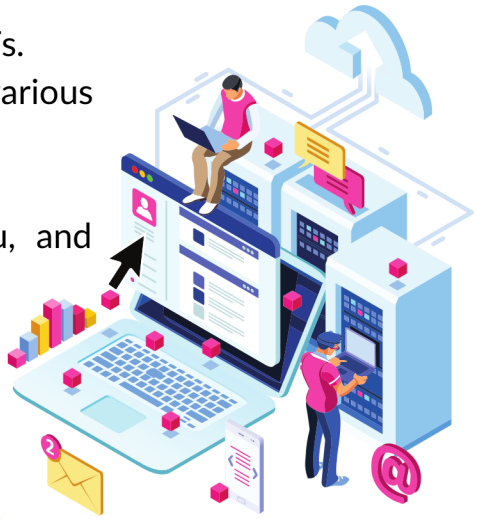
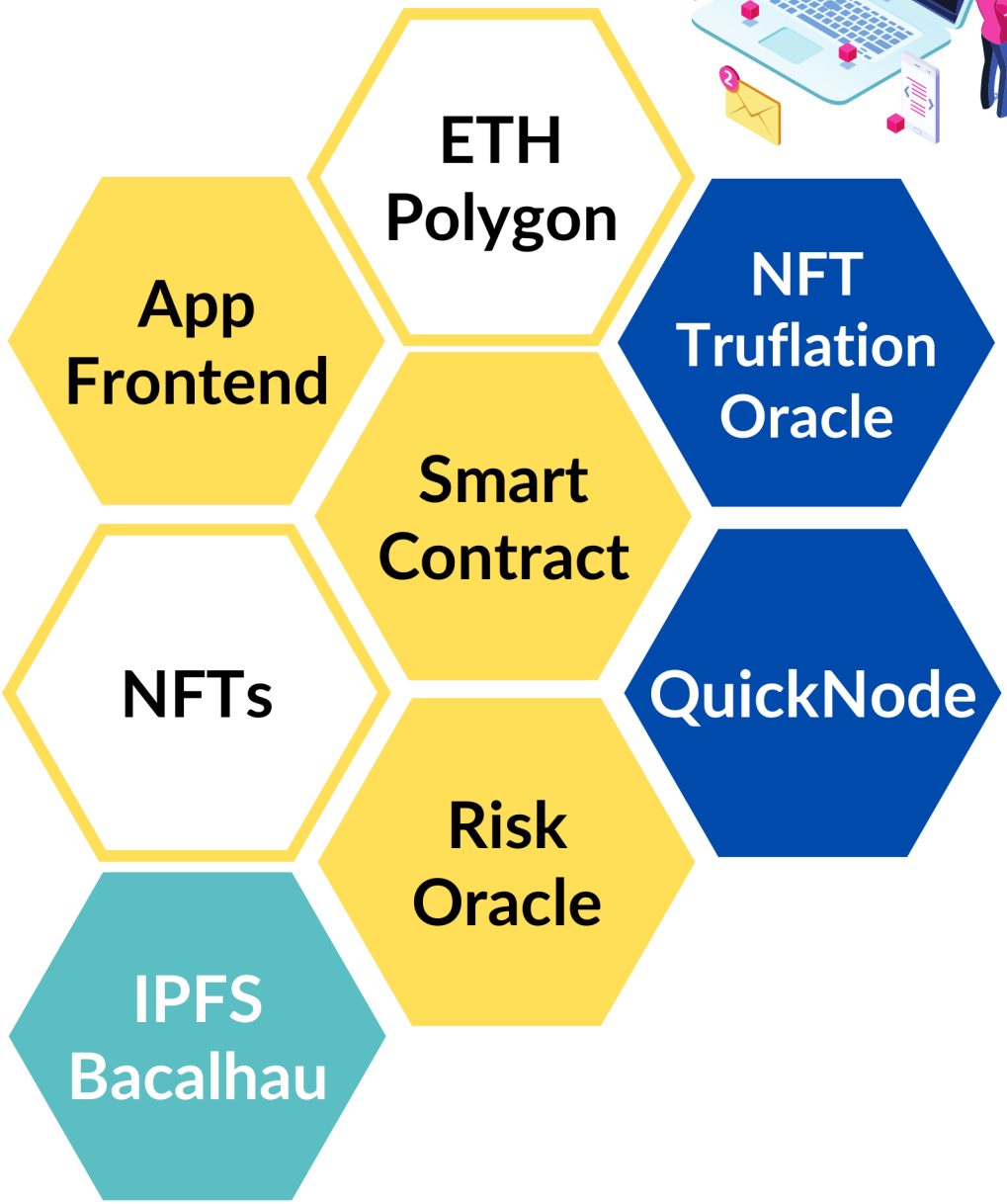
HONEYBEE LOANS

ARCHITECTURE AND SYSTEM DESIGN

HONEYBEE LOANS is a DeFi lending protocol for NFTs.
It offers loans where the collateral are NFTs at various terms.

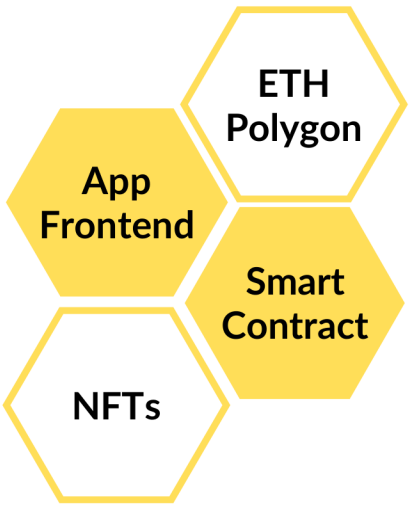
It is built with Chainlink, Truflation, IPFS, Bacalhau, and QuickNode.





SYSTEM COMPONENT OVERVIEW

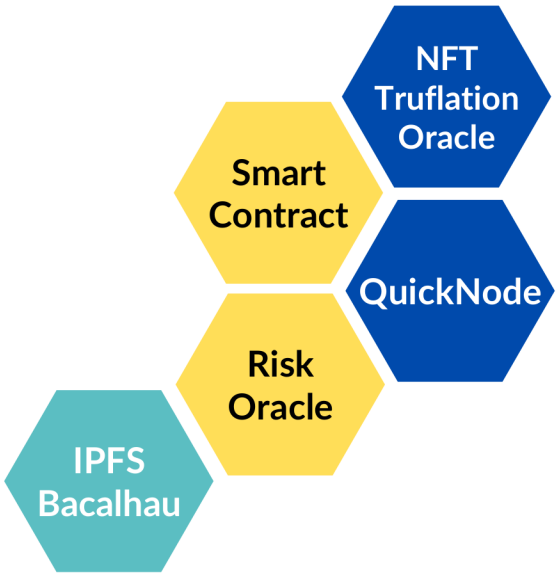
The app frontend is a mobile-friendly React app written in JavaScript. It interacts with the HONEYBEE LOANS smart contract for depositing NFTs and repaying loans. For testnet purposes, loans are denominated in HPT, a prototype ERC20 token. The NFTs of a wallet are evaluated and various loan terms and rates are offered.



For example, users can deposit their Polygon NFTs in exchange for a 30-day loan of 0.1 HPT with an interest rate of 14%. When the loan is paid back in full, the NFT is sent back to the user's wallet.

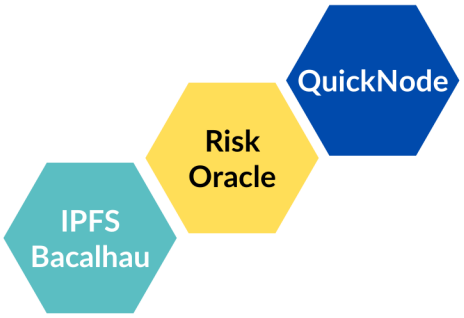
The smart contract has execution functions for depositing the NFT and paying back the loan.

The query functions return data on loan terms and rates based on the wallet address and NFT contract address.



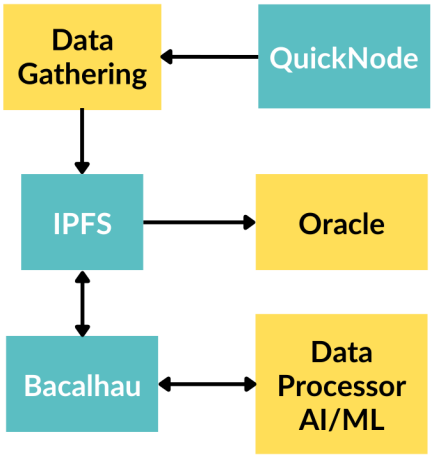
The smart contract relies on an NFT price oracle and a Risk oracle to determine the loan terms and rates that are available. In addition, the loan interest rates factors in the inflation rate from the Truflation oracle. The NFT prices are based on the NFT token id and the NFT contract address.

SYSTEM COMPONENT - THE RISK ORACLE



The Risk oracle is built by our team and evaluates the transaction history of the wallet address and NFT contract address to determine how risky the loan would be.

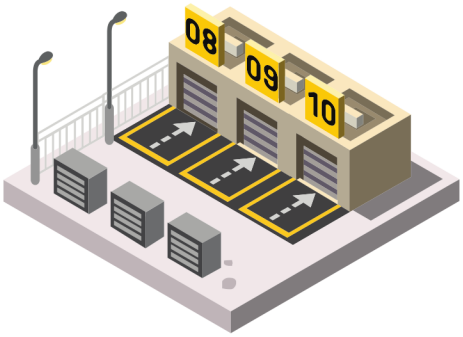
There are three parts to the Risk oracle: data gatherer, data processor, and data provider. The data gatherer searches through wallet and NFT contract transactions, processes them into a data format and then uploads them to IPFS.



The data processor uses TensorFlow AI/ML to train and predict. The results of training and prediction are uploaded to IPFS.

The data provider downloads the data from IPFS, and then factors it into the loan terms and rates that are available.

Data gathering is completed by scraping adnd examining the transactions that have been completed on the wallet. If there is evidence of wallet draining or wash-trading or transactions conducted through suspect exchanges or other flagged wallets, the riskiness of lending to that wallet is higher.

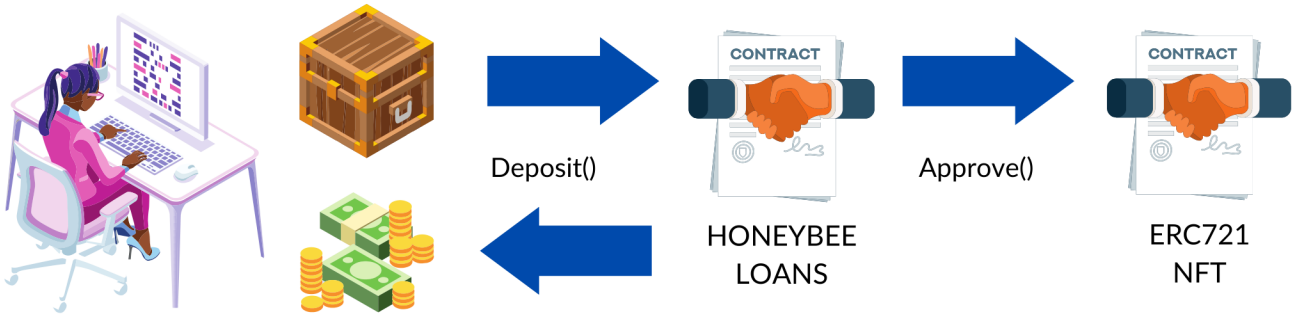


When uploaded to IPFS, the dataset becomes decentralized and is used for training an AI/ML model. If better data and better signals for identifying risky NFTs or wallet addresses becomes available, the AI/ML model can be trained (or re-trained) on those datasets.

The trained AI/ML TensorFlow model provides the data for the Risk oracle that is served to consumers.

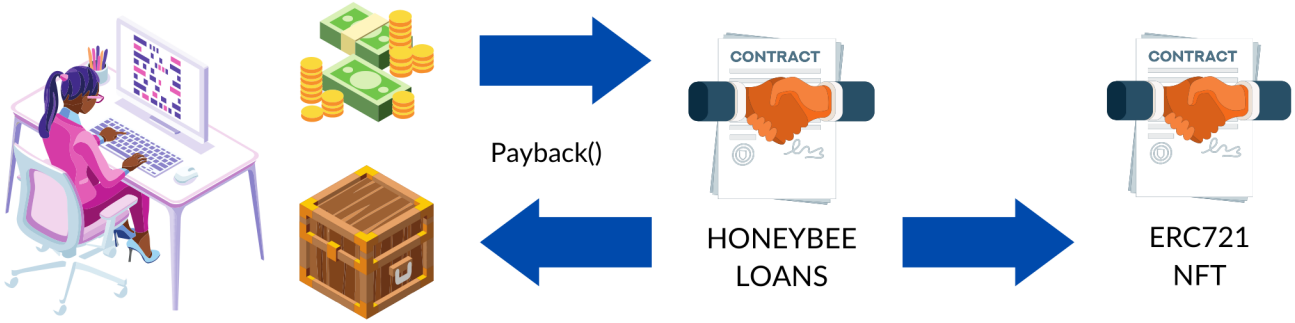
SMART CONTRACT PROCESSES

BORROWING WITH NFT AS COLLATERAL



The borrower selects an NFT to deposit, approves the transfer and receives an amount of tokens in return

REPAYING THE LOAN



The borrower pays back the token with interest bit by bit or in full by the due date and receives the NFT in return

---

Before a loan can be made, the NFT contract must be part of the allowlist of contract addresses and the borrower must be the owner of the NFT they wish to use as collateral. Furthermore, their wallet must have an acceptable risk level, determined by the oracle.

The deployer of the HONEYBEE LOANS contract can control the following variables:

- the minimum and maximum amount that can be loaned
- the interest rate range
- the acceptable risk range

After deploying, these variables can be changed, and can be queried through the public contract getter methods.