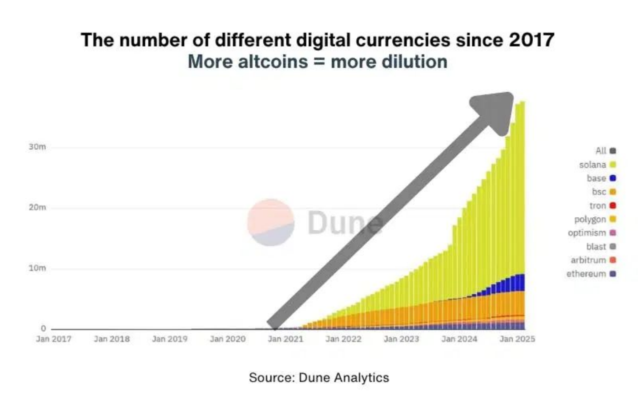
LEEN

Your Token, Your Collateral — Get Stable

Team: Luciano Juvinski, Yifei Ren, Abhishek Bhatnagar, Ayush Jain, Alan Ling

A Market Overflowing



The number of different digital currencies since 2017

More altcoins = more dilution

Source: Dune Analytics

23,000 tokens

A Market Overflowing with Underutilized Tokens

23K

Tokens

Over 23,000 tokens exist

~2%

Listed

only ~2% are listed on major CEXs

300

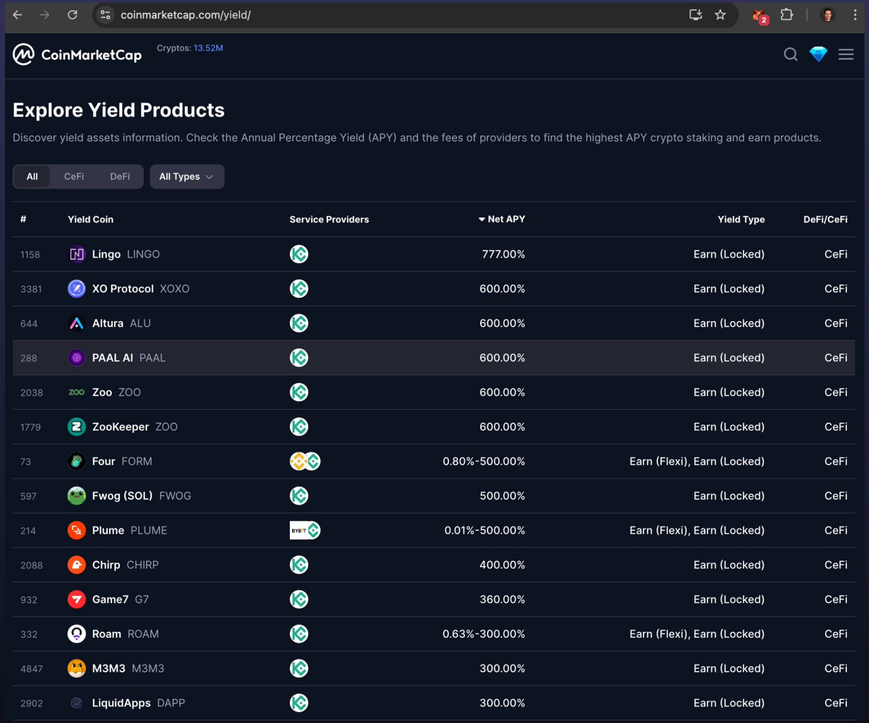
Lending

Lending protocols only support high-liquidity tokens

A Market Overflowing with Underutilized Tokens

Thousands of tokens lack utility beyond speculation or inflationary staking

Project teams struggle to generate real demand without resorting to printing more tokens



Explore Yield Products

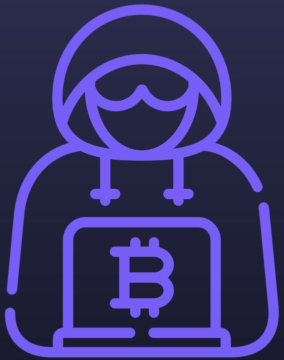
Discover yield assets information. Check the Annual Percentage Yield (APY) and the fees of providers to find the highest APY crypto staking and earn products.

#	Yield Coin	Service Providers	Net APY	Yield Type	DeFi/CeFi
1158	Lingo LINGO		777.00%	Earn (Locked)	CeFi
3381	XO Protocol XOXO		600.00%	Earn (Locked)	CeFi
644	Altura ALU		600.00%	Earn (Locked)	CeFi
288	PAAL AI PAAL		600.00%	Earn (Locked)	CeFi
2038	Zoo ZOO		600.00%	Earn (Locked)	CeFi
1779	ZooKeeper ZOO		600.00%	Earn (Locked)	CeFi
73	Four FORM		0.80%-500.00%	Earn (Flexi), Earn (Locked)	CeFi
597	Fwog (SOL) FWOG		500.00%	Earn (Locked)	CeFi
214	Plume PLUME	BYBIT	0.01%-500.00%	Earn (Flexi), Earn (Locked)	CeFi
2088	Chirp CHIRP		400.00%	Earn (Locked)	CeFi
932	Game7 G7		360.00%	Earn (Locked)	CeFi
332	Roam ROAM		0.63%-300.00%	Earn (Flexi), Earn (Locked)	CeFi
4847	M3M3 M3M3		300.00%	Earn (Locked)	CeFi
2902	LiquidApps DAPP		300.00%	Earn (Locked)	CeFi

Sustainable Utility Through Native Credit Markets

The Solution for Token Issuers / Project Owners

- LEEN enables projects to create their own lending pools backed by stablecoins
- Fully configurable: LTV ratio, interest rate, liquidation threshold, price band, oracle, duration
- Use ICO treasury capital to fund the pool, creating demand without inflation
- Interest collected from loans can be automatically reinvested to grow the pool
- The protocol becomes a self-sustaining liquidity engine



LIQUIDITY LAYER LANDSCAPE

	Swap	Lending	Stake
High Liquidity Tokens (<300)	CENTRALIZED EXCHANGE BINANCE kraken coinbase OKX KuCoin	TOP 100 TOKENS AAVE Frax Compound venus Maker	Lido yearn Beefy Balancer Curve Bactor
Other 20k Tokens	DECENTRALIZED EXCHANGE UNISWAP SushiSwap Balancer PancakeSwap	ANY TOKEN LEEN Your Token, Your Collateral— Get Stable	

Borrow Without Selling — Stay Exposed to the Upside

The Solution for Investors and Token Holders

Token holders can borrow stablecoins using their tokens as collateral

Collateral

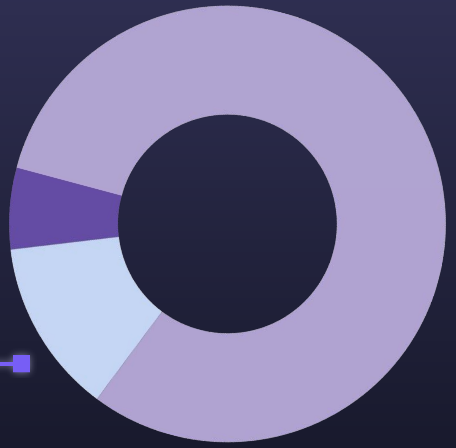
No need to sell their tokens — they keep exposure to price appreciation

Exposure

Enables reinvestment strategies using borrowed stablecoins

Reinvest

Economic Model



1%

Loan fee: fixed protocol commission

5%

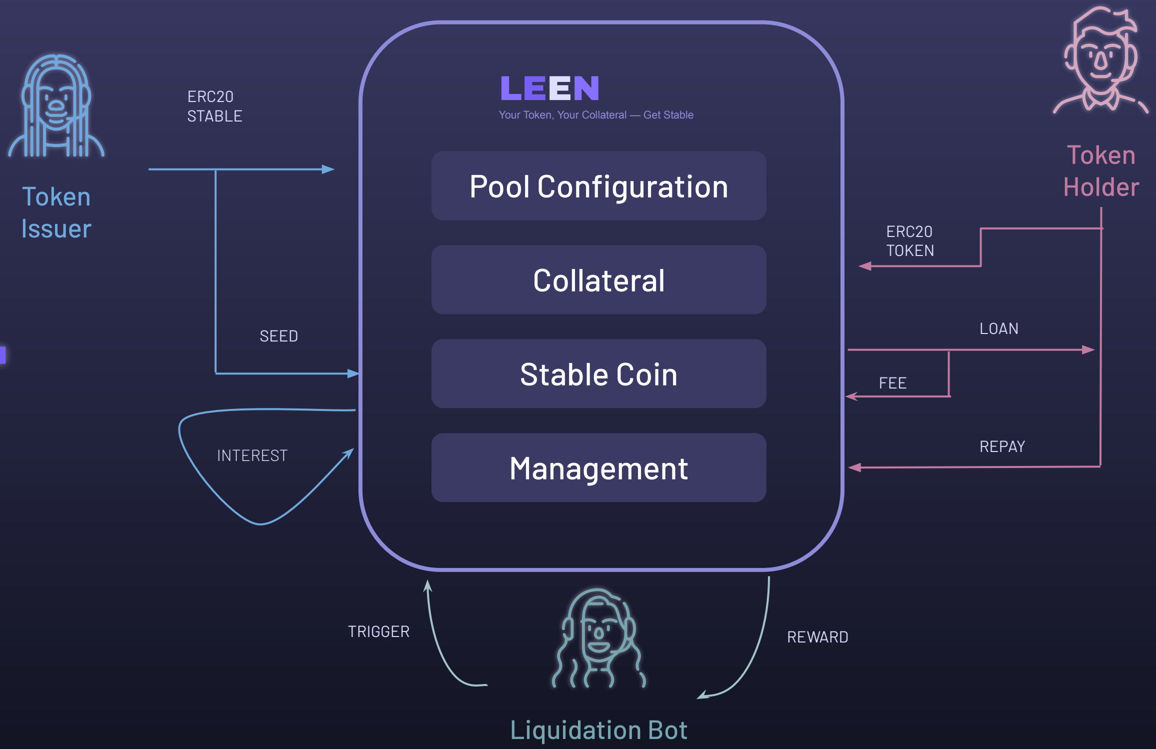
Liquidation fee: split with executor bots

94%

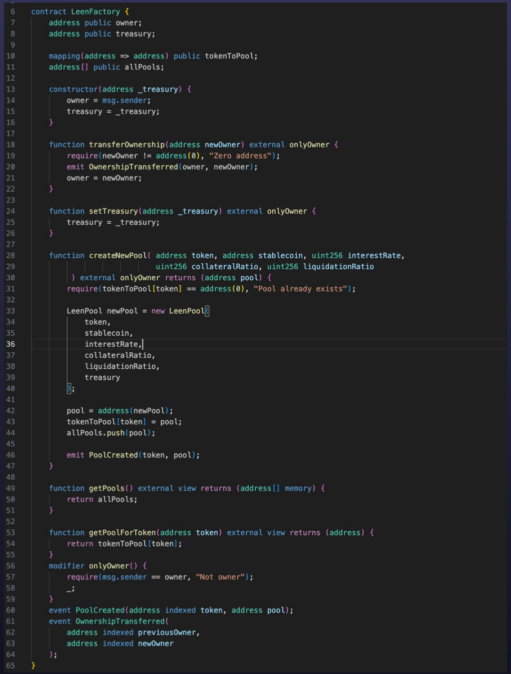
Issuers receive interest from borrowers

The interest rate range is 3-10%

Architecture



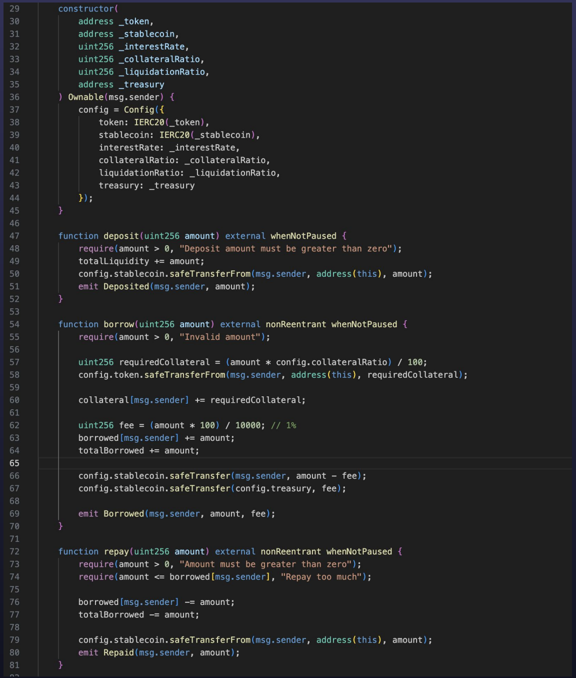
Smart contract



```solidity
contract LeenFactory {
    address public owner;
    address public treasury;

    mapping(address => address) public tokenToPool;
    address[] public allPools;

    constructor(address _treasury) {
        owner = msg.sender;
        treasury = _treasury;
    }

    function transferOwnership(address newOwner) external onlyOwner {
        require(newOwner != address(0), "Zero address");
        emit OwnershipTransferred(owner, newOwner);
        owner = newOwner;
    }

    function setTreasury(address _treasury) external onlyOwner {
        treasury = _treasury;
    }

    function createNewPool( address token, address stablecoin, uint256 interestRate,
                            uint256 collateralRatio, uint256 liquidationRatio
     ) external onlyOwner returns (address pool) {
        require(tokenToPool[token] == address(0), "Pool already exists");

        LeenPool newPool = new LeenPool(
            token,
            stablecoin,
            interestRate,
            collateralRatio,
            liquidationRatio,
            treasury
        );

        pool = address(newPool);
        tokenToPool[token] = pool;
        allPools.push(pool);

        emit PoolCreated(token, pool);
    }

    function getPools() external view returns (address[] memory) {
        return allPools;
    }

    function getPoolForToken(address token) external view returns (address) {
        return tokenToPool[token];
    }
    modifier onlyOwner() {
        require(msg.sender == owner, "Not owner");
        _;
    }
    event PoolCreated(address indexed token, address pool);
    event OwnershipTransferred(
        address indexed previousOwner,
        address indexed newOwner
    );
}
```

Factory ✓

https://sepolia.etherscan.io/address/0xaCE3302564cbA291Dc72BD5D55B5434Cb801d845#code

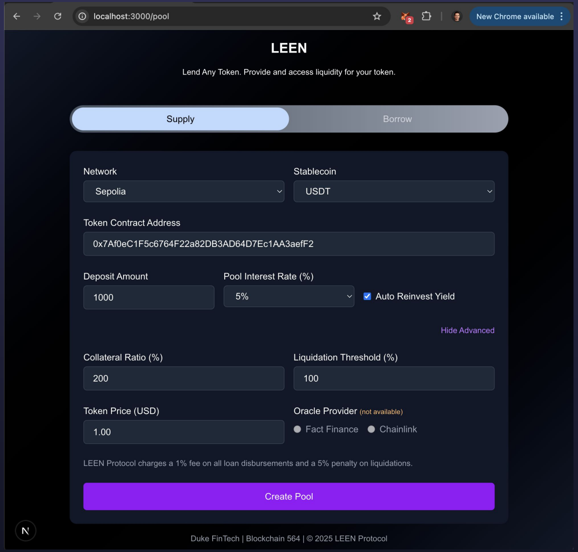


```solidity
    constructor(
        address _token,
        address _stablecoin,
        uint256 _interestRate,
        uint256 _collateralRatio,
        uint256 _liquidationRatio,
        address _treasury
    ) Ownable(msg.sender) {
        config = Config({
            token: IERC20(_token),
            stablecoin: IERC20(_stablecoin),
            interestRate: _interestRate,
            collateralRatio: _collateralRatio,
            liquidationRatio: _liquidationRatio,
            treasury: _treasury
        });
    }

    function deposit(uint256 amount) external whenNotPaused {
        require(amount > 0, "Deposit amount must be greater than zero");
        totalLiquidity += amount;
        config.stablecoin.safeTransferFrom(msg.sender, address(this), amount);
        emit Deposited(msg.sender, amount);
    }

    function borrow(uint256 amount) external nonReentrant whenNotPaused {
        require(amount > 0, "Invalid amount");

        uint256 requiredCollateral = (amount * config.collateralRatio) / 100;
        config.token.safeTransferFrom(msg.sender, address(this), requiredCollateral);

        collateral[msg.sender] += requiredCollateral;

        uint256 fee = (amount * 100) / 10000; // 1%
        borrowed[msg.sender] += amount;
        totalBorrowed += amount;

        config.stablecoin.safeTransfer(msg.sender, amount - fee);
        config.stablecoin.safeTransfer(config.treasury, fee);

        emit Borrowed(msg.sender, amount, fee);
    }

    function repay(uint256 amount) external nonReentrant whenNotPaused {
        require(amount > 0, "Amount must be greater than zero");
        require(amount <= borrowed[msg.sender], "Repay too much");

        borrowed[msg.sender] -= amount;
        totalBorrowed -= amount;

        config.stablecoin.safeTransferFrom(msg.sender, address(this), amount);
        emit Repaid(msg.sender, amount);
    }
```

Pool ✓

https://sepolia.etherscan.io/address/0xE5d6370Dd99cCfa415d5c480970886799D3DB510#code

dApp



localhost:3000/pool

New Chrome available

LEEN

Lend Any Token. Provide and access liquidity for your token.

Supply | Borrow

Network: Sepolia

Stablecoin: USDT

Token Contract Address: 0x7Af0eC1F5c6764F22a82DB3AD64D7Ec1AA3aefF2

Deposit Amount: 1000

Pool Interest Rate (%): 5%

☑ Auto Reinvest Yield

Hide Advanced

Collateral Ratio (%): 200

Liquidation Threshold (%): 100

Token Price (USD): 1.00

Oracle Provider (not available)

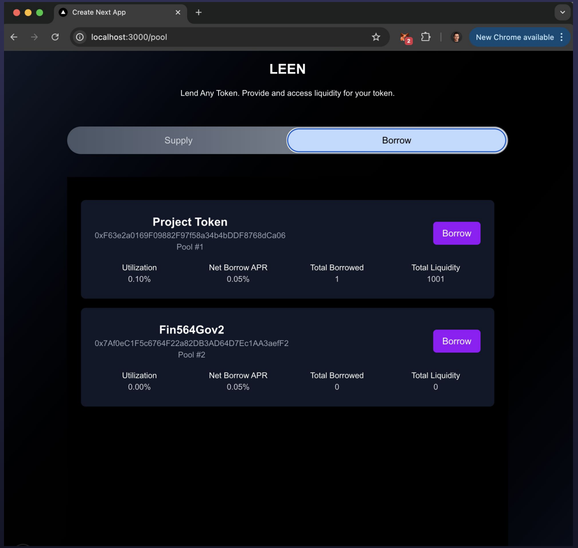
○ Fact Finance ○ Chainlink

LEEN Protocol charges a 1% fee on all loan disbursements and a 5% penalty on liquidations.

Create Pool

Duke FinTech | Blockchain 564 | © 2025 LEEN Protocol

Pool Creation



Create Next App

localhost:3000/pool

New Chrome available

LEEN

Lend Any Token. Provide and access liquidity for your token.

Supply | Borrow

Project Token

0xF63e2a0169F09882F97f58a34b4bDDF8768dCa06

Pool #1

Borrow

Utilization	Net Borrow APR	Total Borrowed	Total Liquidity
0.10%	0.05%	1	1001

Fin564Gov2

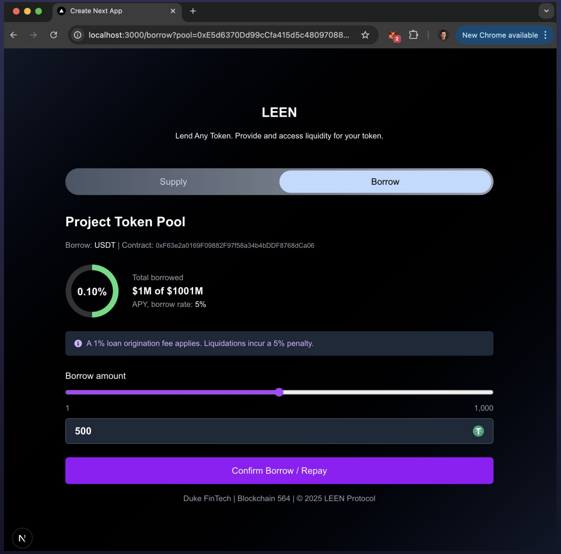
0x7Af0eC1F5c6764F22a82DB3AD64D7Ec1AA3aefF2

Pool #2

Borrow

Utilization	Net Borrow APR	Total Borrowed	Total Liquidity
0.00%	0.05%	0	0

List



Create Next App

localhost:3000/borrow?pool=0xE5d6370Dd99cCfa415d5c48097088...

New Chrome available

LEEN

Lend Any Token. Provide and access liquidity for your token.

Supply | Borrow

Project Token Pool

Borrow: USDT | Contract: 0xF63e2a0169F09882F97f58a34b4bDDF8768dCa06

0.10%

Total borrowed

$1M of $1001M

APY, borrow rate: 5%

A 1% loan origination fee applies. Liquidations incur a 5% penalty.

Borrow amount

1 — 1,000

500

Confirm Borrow / Repay

Duke FinTech | Blockchain 564 | © 2025 LEEN Protocol

Borrow Page

http://vcm-47408.vm.duke.edu/leen

> “If your token can’t pump, at least let it earn. Deploy a pool. Print some yield. Stay LEEN”.

—Duke FinTech 564

Thank you!