P I N J A M

Pinjam Protocol Whitepaper V1.0

shawn@pinjamlabs.com

January 2023

Lending & Borrowing protocol based on Aave V2 and Geist Finance.
With our unique tool, we add to the traditional Pooled Lending Model, the option for unused depositor funds to be staked on selected blue chip protocols, ensuring better rewards and 100% capital efficiency.

Abstract

The Pinjam (Pronounced: Pin-Jump) Protocol is the natural evolution of economic systems designed on the blockchain. It introduces a fully on-chain Fractional Reserve Lending protocol. This document explains why such a protocol is needed for lenders & borrowers, how it benefits the DeFi ecosystem and how it will accrue value to token holders.

Content Outline

1. Introduction

The Pinjam Protocol is a money market protocol designed with the objective of achieving 100% Capital Productivity by putting unborrowed capital into other blue chip protocols of the same blockchain via Pinjam Vaults, generating more delta neutral yield for lenders. The core protocol is architected and designed based on AaveV2, Yearn, and Beefy, combined into the Pinjam Protocol achieving the result of the first fully on-chain fractional reserve money market.

1.1 Problems

The development of base lending protocols such as Aave, Compound, and BenQi has brought in significant interest and exposed the first tens of billions of dollars to understand the benefits of Decentralised Lending, such as, acquiring loans without the need of a long credit check process, transparently understand how interest rates are calculated, gain leverage exposure without the use of CeFi solutions, etc. These are the benefits of providing an on-chain money market, and they work wonderfully well.

However, the problem is in the incentives of borrowers and lenders and the fact that they are not aligned. To understand this, one first needs to understand that supply & borrow rates are determined by the utilization of assets in a reserve pool. The more an asset is being borrowed the higher the cost of borrowing thus lenders earn more from a higher supply rate.

This means borrowers want more liquidity in order to borrow cheaply, while lenders want less liquidity so that borrowing is more expensive thus giving lenders more yield.

This unaligned incentive structure is most visible in overcollateralized money markets today because borrowers want lower borrow rates and to achieve that lenders need to deposit more liquidity, but lenders do not want to deposit more liquidity because the supply rates are too low because borrowers are only willing to pay x amount in borrowing costs before it becomes too expensive for them to want to borrow more.
This un-aligned incentive structure stems from the issue of Idle Liquidity, which is defined as so:

Idle Liquidity = *Liquidity/Capital Sitting Unborrowed in Reserve Not Generating Yield for Lenders*

And is derived as so:

$$Idle\ Liquidity\ =\ 1\ -\ \frac{Total\ Borrowed}{Total\ Supplied}$$

In a money market, Idle Liquidity appears when there is low or insufficient borrow demand, and because interest rates are determined from the utilization/borrow rate, the higher the idle liquidity, the lower the yield earned by lenders but the cheaper it is to borrow.

This means borrowers want more Idle Liquidity while lenders want less Idle Liquidity, resulting in the mismatch of incentives.

Take the table below for example to see the current percentage of idle liquidity in base lending protocols.

Protocol	Total Supplied	Total Borrowed	Idle Liquidity
Aave V2 (Ethereum)	$5.57B	$1.76B	68%
Aave V3 (Avalanche)	$497.76M	$201.96M	59%
Compound V2	$2.272B	$0.716B	68%
Compound V3	$284.34M	$116.12M	59%
BenQi	$163.44M	$37.38M	77%
Venus	$1.39B	$0.496B	64%
Euler	$435.37M	$203.18M	53%

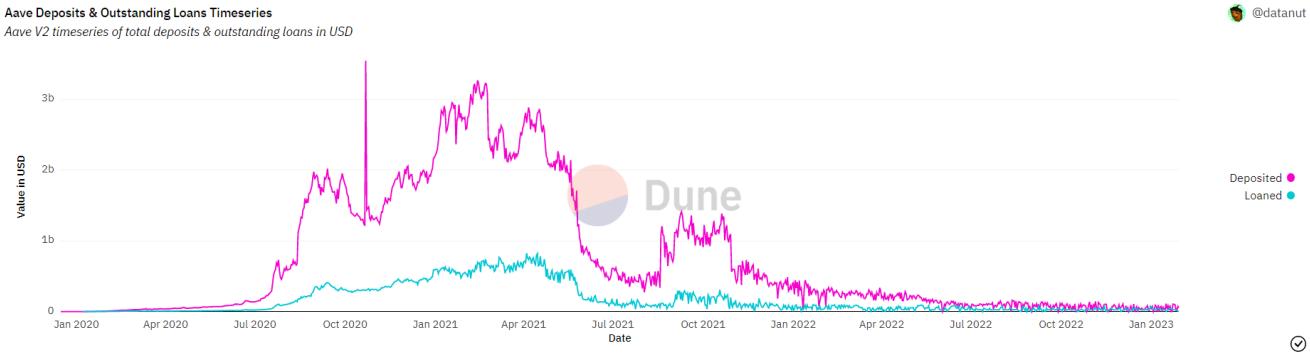
Source: Taken from open data on protocol's website

***Average Idle Liquidity** = 64%*

As shown above, across all major lending protocols today, many of them are sitting, on average, above 60% in idle liquidity.

This observation isn't isolated to appear only in bear market conditions, looking at the historical data on total deposits vs total borrowed in Aave V2 on ethereum it shows that even in peak bull run environment, there was significant idle liquidity.

Take a look at the historical data below:



Aave Deposits & Outstanding Loans Timeseries [5]

The above chart represents the total Deposits and Outstanding loans on Aave V2 on the Ethereum network. It shows that during the whole bullrun of 2021, there was, on average, about 60% of deposits sitting idle being unproductive and unborrowed, similar to the current environment we are now in 2023. More historical data like this can be found on BenQi as they are very open with their data and have a historical chart showing the total deposits vs total borrowed.

To take it further, due to the nature of over-collateralization, which means for every $1 a lender deposits, that same lender can only borrow up to a maximum of $0.80 cents. This means that even if a lending protocol reaches maximum utilization rate, there will always be some liquidity remaining idle and not being productive to lenders.

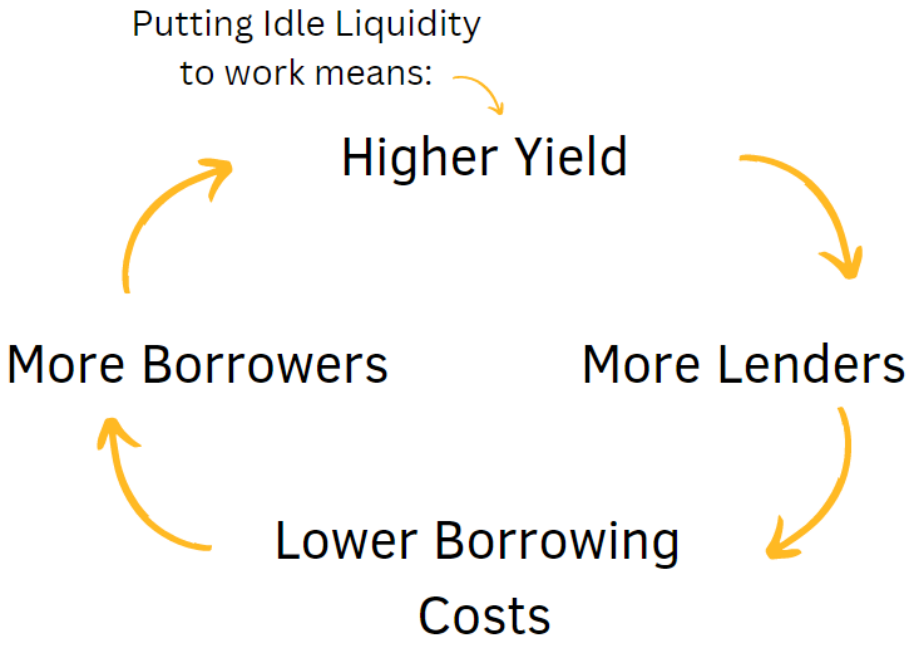
The problem of Idle Liquidity results in 3 main problems:

1. For every $1 a lender deposits, only $0.40 cents is productive earning yield for the lender.
2. Inefficient value accrual to token holders.
3. Liquidity Fragmentation across the decentralized finance ecosystem.

1.2 Solution

The Pinjam Protocol will reduce the impact of the mismatch incentives between lenders and borrowers. It does this by taking a concept similar to the Iron Bank of Protocol-to-Protocol lending by lending Idle Liquidity to other blue-chip protocols which will make unborrowed/unproductive capital productive, generating more yield for lenders.

Lenders ultimately want higher yield, which can only be achieved by having less liquidity in the pool while having more borrowing activity. By putting Idle Liquidity to work, we reduce the negative impact of these mismatch incentives because it no longer matters how much liquidity is in the pool being borrowed, whether there is more or less borrowing activity, 100% of the deposits will still be productive, consistently generating yield, no matter the borrowing activity. Doing this will result in what we are coining, the Yield Flywheel.



Yield Flywheel

By putting Idle Liquidity to work, the Pinjam Protocol increases the supply rate earned by lenders, therefore incentivizing more lenders to deposit increasing overall liquidity which then results in lower borrowing costs, which will then allow more borrowers to borrow at a reasonable borrow rate.

This solution makes the Pinjam Protocol the first fully on-chain Fractional Reserve Lending Solution and it results in 4 main benefits:

1. Every $1 a lender deposits will always be at work, and productivity will be maximized via the mix of borrowing & farming activities.

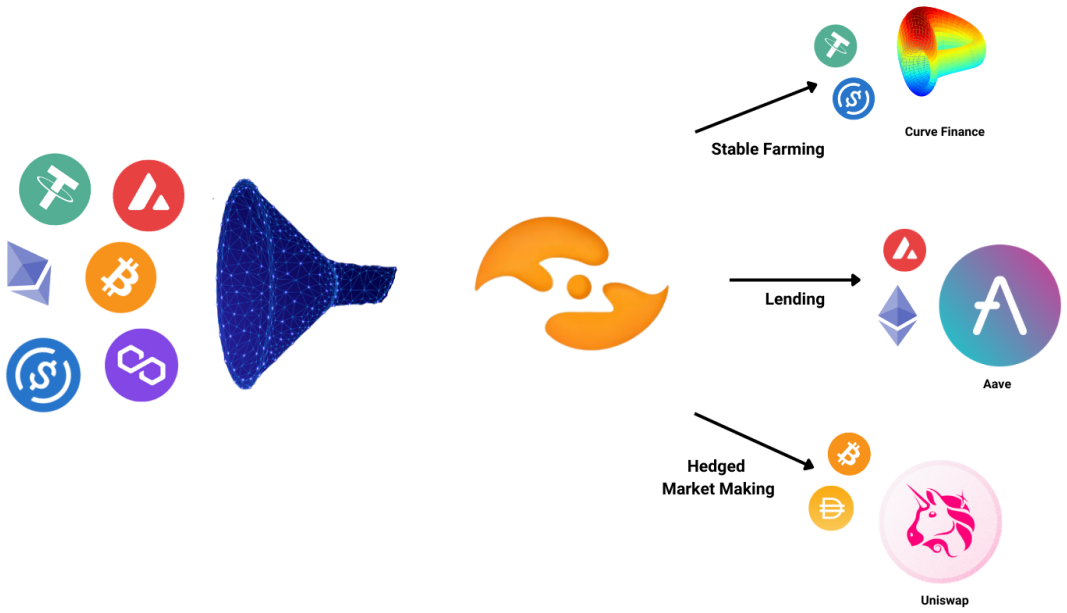
2. The Pinjam Vault is another form of revenue for the Pinjam protocol, resulting in higher value accrual to token holders. 50% of the Pinjam Vault revenue will be distributed to token stakers & lockers.

3. Decentralized Finance will stand to benefit from higher liquidity distributed by the Pinjam Vault, resulting in lower slippage, lower cost of borrowing, and reducing liquidity fragmentation across different protocols.

4. Taking DeFi closer to TradFi efficiency while eliminating the traditional risk of hidden counterparty risks through transparent finance by being fully on-chain.

1.3 Yield Sources

It is important to note that the Pinjam Protocol can only exist today due to some level of maturity in the DeFi ecosystem. DeFi is effectively 5 years old, and there are several trusted protocols, or protocols that can be considered blue-chip such as Aave, Compound, Uniswap, Curve, and several more.



Pinjam as a Liquidity Aggregator

The Pinjam Protocol will route Idle Liquidity through highly trusted and battle-tested protocols to earn yield for lenders safely. It is not intended to be a degenerate farm promising 10% - 20% yield as that is more than likely unsustainable and is susceptible to ponzinomic collapses.

For the current V1 implementation of the Pinjam Protocol, it will only route Idle Liquidity through base layer lending protocols such as Aave and BenQi as it will be first launched on the Kavan and Mantle networks.

V2 improvements could include routing Idle Liquidity through AMMs such as Curve, TraderJoe, and Beefy.

1.4 Quantifying The Yield

This section will cover how much more yield Pinjam will generate by only depositing to a single base lending protocol, Aave.

Given the hypothetical nature of this experiment, an assumption will be made that both supply and borrow rates will be similar on Pinjam and Aave due to market forces and yield arbitrage opportunities.

Let's take 3 assets on Aave as an example, USDT, DAI, and ETH. Currently on 30th January the supply rates on Aave V2 on the Ethereum network are as follows:

Asset	**Supply Rate**	**Idle Liquidity**
USDT	2.78%	20%
DAI	1.64%	40%
ETH	1.25%	54%

Supply Rate & Idle Liquidity on Aave V2 Ethereum, on 30th January 2023

The following table represents the total supply rate that would be earned by lenders on Pinjam:

Asset	**Supply Rate**	**Vault Rate**	**Total Supply Rate**	**Improvement**
USDT	2.78%	0.556%	3.336%	20%
DAI	1.64%	0.656%	2.296%	40%
ETH	1.25%	0.675%	1.925%	54%

The Estimated Total Supply Rate Earned By Lenders on Pinjam with Aave

As expected, the less idle liquidity on Aave, the less in improvement in supply rate offered by Pinjam.

The above example is a simple representation on how much the Pinjam protocol will be able to improve yield by simply using Aave as the sole yield source. If we take that same example and instead redirect Idle Liquidity to Yearn where they average 7% yield per annum [1] the total supply rate earned by lenders will be as so:

Asset	**Supply Rate**	**Vault Rate**	**Total Supply Rate**	**Improvement**
USDT	2.78%	1.4%	4.18%	50%

DAI	1.64%	2.8%	4.44%	170%
ETH	1.25%	3.78%	5.03%	300%

The Estimated Total Supply Rate Earned By Lenders on Pinjam with Yearn

By redirecting the yield source from a blue chip protocol such as Aave to another blue chip protocol like Yearn, we can see a 3x improvement in the supply rate earned by lenders.

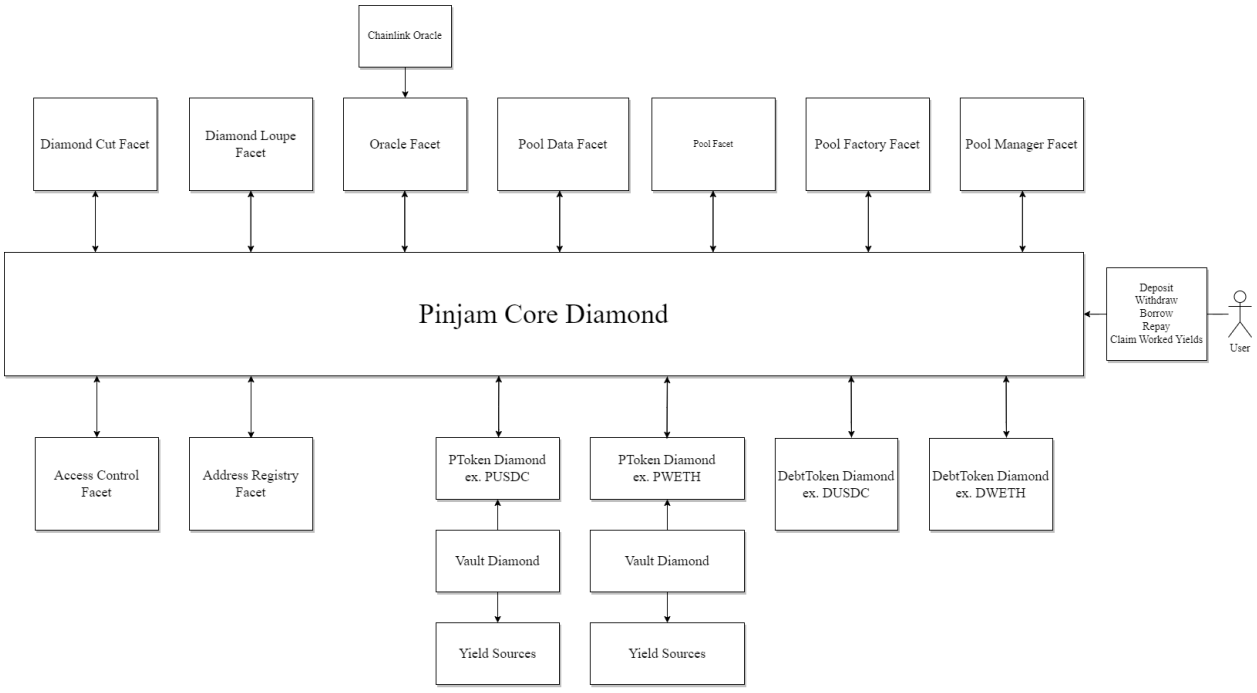
This is the opportunity of a protocol like Pinjam, as DeFi matures and newer protocols release more real use cases with yield-generating opportunities, lenders in Pinjam will not need to do anything but simply watch their capital auto-compound and collect the yield.

2. Protocol Architecture

2.1 High Overview

The Pinjam Protocol follows a similar architecture introduced by Aave V1 and V2, where the heart of the protocol, sits the lending pool which a user interacts with.

However that architecture has been rebuilt using the EIP:2535 standard and this is how it looks like now:



Pinjam Smart Contract EIP:2535 Diamond Architecture

With the Pinjam Core Diamond in the center of all operations, functional facets are added to fulfill the needs of the protocol.

The redesign of the smart contract architecture resulted in a single core contract address where most of the functions are executed, for both the users and admin. This single contract address is able to essentially provide the functionality of multiple contracts (facets) that are independent from each other but is able to share components such as libraries, state variables, and internal functions. A single contract address has also eased deployment, testing and integration efforts.

The key factors in deciding to rebuild and use the EIP:2535 is so that we could manage upgrades in a modular fashion and reuse facets where we could. It also enabled and solved the limitation of 24kb of

maximum contract size. Data management was also improved due to the more systematic approach to isolating different functions and the usage of data structures within the diamond facets.

While the EIP:2535 standard has brought along many new improvements in smart contract architecture it introduces a new and incredibly high vulnerability threat, which is that the owner EOA can upgrade the facet, or create a new facet which can drain all the funds. Even if the owner is benevolent, the risk of the private key being compromised is a big deal. To resolve this security vulnerability the owner of the *Pinjam Core Diamond* will have its ownership transferred to a timelock contract which will be managed by a multisig. This will be dealt with after the Mantle launch.

2.1.1 Facets

2.1.1.1 Access Control Facet

The access control facet replaces the default ownership facet from EIP:2535. This access control facet now adds a new role which is the admin which is required by some functions to be executed.

Ownership of the diamond can be changed here by the current owner.

2.1.1.2 Address Registry Facet

The address registry facet lists all addresses of the current implementation of facets within the Diamond.

2.1.1.3 Diamond Cut Facet

The diamond cut facet is used to add, replace and remove functions from the main diamond. This function can only be called by the owner.

2.1.1.4 Diamond Loupe Facet

The diamond loupe functions can be used to show all functions used by a diamond, to find out what functions a diamond has before calling it. To be used by query services like https://louper.dev/ to retrieve a Diamond's implementation.

2.1.1.5 Oracle Facet

The oracle facet maps underlying assets to an oracle source of the assets, this mapping can be added and updated by an admin. The current main oracle source in use is Chainlink.

2.1.1.6 Pool Data Facet

The pool data facet serves as a data center for users and the main diamond own internal used to query specific on-chain data related to Pinjam. Such data involves querying the borrow and supply rates, the total liquidity and debt of an asset, fetching the address of the PToken and DebtToken of an underlying asset, to retrieve user data, which includes the health factor, and average loan to value.

2.1.1.7 Pool Facet

The pool facet will have the function that user's will mostly interact with such as deposit, withdraw, borrow, repay, and claimWorkedYields. The functions are as below:

$deposit(address\ underlyingAsset,\ uint256\ amount,\ address\ to,\ bool\ depositToVault)\ external;$
$withdraw(address\ underlyingAsset,\ uint256\ amount,\ address\ to)\ external;$
$borrow(address\ underlyingAsset,\ uint256\ amount,\ address\ onBehalfOf)\ external;$
$repay(address\ underlyingAsset,\ uint256\ amount,\ address\ to)\ external;$
$claimWorkedYields(address\ underlyingAsset,\ address\ to)\ external;$

2.1.1.8 Pool Factory Facet

The pool factory is used by the admin to create a new lending and borrowing pool based on an underlying asset. This factory will deploy a new PToken and DebtToken Diamond accordingly.

2.1.1.9 Pool Manager Facet

The pool manager facet is used to change pool settings and parameters. From settings such as supply and debt limit, to changing the current yield source being used.

2.2 Basic Mechanics

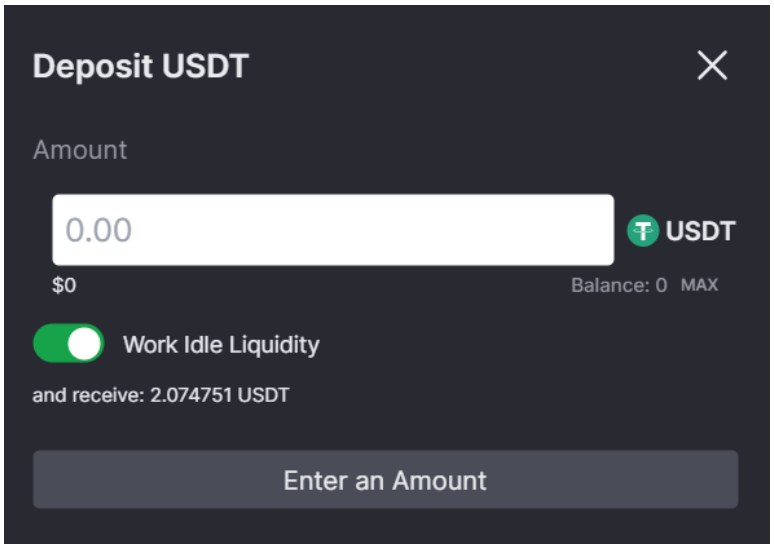
This section will cover all the basic mechanics expected from a money market and how the Pinjam Vaults differ with existing liquidity protocols.

2.2.1 Supplying Assets & Maintaining Desired Capital Efficiency

The Pinjam Protocol will maintain a ***Maximum Desired Capital Efficiency*** for each asset, we only want less capital efficiency on more expensive networks like Ethereum where the Pinjam Protocol constantly depositing and withdrawing assets from the vaults becomes more expensive for users. On cheaper networks like Kava, Mantle and Fantom, it is more acceptable to have 100% capital efficiency as gas costs are significantly cheaper.

When a lender deposits money into the protocol without the intention of borrowing, that lender is contributing to Idle Liquidity. In a normal money market this can be recognized as ***toxic liquidity*** to other lenders, as this will reduce the total supply rate earned by other lenders.

To avoid this, the Pinjam Protocol will incentivize depositors to help contribute and maintain the capital efficiency by allowing new depositors the option to put all Idle Liquidity to work when they deposit.
Doing so, they will receive 1% of the total rewards accrued in this vault since the previous deposit.



https://staging.pinjamlabs.com

2.2.2 Borrowing, Repaying, Withdrawing & Liquidation of Assets

Borrowing, repaying, withdrawing and liquidation of assets is similar to Aave V2, and its Loan to Value and Health Factor implementation. Once a user's Health Factor drops below 1, their account will be up for liquidation nor will they be allowed to withdraw any assets until the debt has been repaid. Further details and math can be referred to the Aave V1 whitepaper under section 1.1 and Aave V2 whitepaper under section 3.1 and 3.2. [2] [3]

The only thing to note about liquidations on Pinjam is that it will remain close sourced and only the team behind the Pinjam Protocol will be allowed to liquidate user assets. This is practiced among CEX's where they have their own internal liquidation engine. The decision for this is purely to maximize value accrual to Pinjam Token Holders, it simply does not make sense to allow 3rd party liquidators to profit out of liquidating users, once the Pinjam token is live, liquidation will be another form of revenue for the protocol and for the stakers of the Pinjam token.

2.2.3 Rebalancing Vault Assets to different strategies

Rebalancing vault assets is an admin only function and is called as such:

rebalance(uint256 fromStrategyIndex, uint256 toStrategyIndex, uint256 amount)

This function will rebalance assets from a strategy to another strategy based on the index and amount input, and will be called if there is a strategy that can bring in higher yield or if the Pinjam Protocol is trying to reduce exposure or hedge its risk with another yield source.

2.2.4 If 100% of Liquidity is at work, how would users withdraw, borrow or get liquidated?

When a user tries to withdraw or borrow, but there is no Idle Liquidity in the reserve pool, the Pinjam Protocol will automatically withdraw from the Pinjam Vault, and transfer it directly to the user in the same transaction.

2.3 Flash Loans

Flash loans will not be supported on the Pinjam protocol for two reasons. Firstly, flash loans are simply a race to the bottom made evident by Euler Finance where there is a 0% flash loan fee, therefore not really bringing any value to the protocol.

The second reason is that Flash Loans are only made possible due to the Idle Liquidity sitting in a smart contract. There will be little to no Idle Liquidity sitting in the Pinjam Protocol, therefore to conduct a flash loan, the protocol will first have to withdraw the assets from the vault, transfer it to the developer's smart contract, which will add-on more gas fees making it less competitive for MEV bots and reduces profitability, making Pinjam a less than ideal source for flash loans.

Ultimately base layer protocols like Aave serve as the best form for Flash Loans anyway since the Pinjam Vault will lend out its Idle Liquidity to these protocols.

2.4. Interest Rates/Rewards Calc

This section dives into how the Pinjam Vault affects the supply and borrow interest rate.

2.4.1 Borrow Interest

The core variable borrow interest rate follows the same model in Aave V1 and remains unchanged.

The asset will take in a Interest Rate Slope 1 (*slope1*) & Interest Rate Slope 2 (*slope2*), and is calculated based on the Current Utilization Rate (U) against the Optimal Utilization Rate ($U_{optimal}$).

Current Utilization Rate (U) can be derived as:

$$U = \frac{Total\ Debt}{Total\ Supplied\ Liquidity}$$

If Current Utilization Rate (U) $<$ Optimal Utilization Rate ($U_{optimal}$):

$$Borrow\ Rate = \frac{U}{U_{optimal}} \times slope1$$

If Current Utilization Rate (U) $>$ Optimal Utilization Rate ($U_{optimal}$):

$$Borrow\ Rate = Slope1 + \left(\frac{U - U_{optimal}}{1 - U_{optimal}} \times slope2\right)$$

In short, the borrow rate will remain unaffected by the vault and is determined by the utilization rate of the asset pool.

For further details and explanation on the borrow rate, refer to Aave's V1 whitepaper under section 2.5 Interest Rate Strategy. [2]

2.4.2 Supply Interest

The supply interest is derived from 2 avenues:

1. The Borrow Utilization Rate
2. The Pinjam Vault Effective Rate

The supply rate from borrowing activity is derived as so:

$$Supply\ Rate\ =\ Borrow\ Rate\ \times\ Utilization\ Rate$$

The Pinjam Vault Effective Rate should be derived off-chain and the calculation of it is as so:

$$i = number\ of\ yield\ sources$$
$$Total\ Supply\ Rate\ =\ Supply\ Rate\ +\ (\sum_{n=0}^{i} Yield\ Source\ Rate(i) \times \%\ of\ funds\ at\ work\ )$$

2.5 Vault Rewards

The rewards generated by the vault are inconsistent and cannot be reliably calculated by using block timestamp or block number, to overcome this, the Pool Facet will have an internal accounting of the amount of Idle Liquidity that has been worked, which can be labeled as *Amount Worked*.

To calculate the current *Total Vault Rewards*, one will have to call the balance function on the Vault which will return the *current* total balance being worked and yield generated which will allow the protocol to derive the total current vault rewards as so:

$$Total\ Current\ Vault\ Rewards\ =\ Total\ Vault\ Balance\ -\ Amount\ Worked$$

The farming rewards generated by the Pinjam Vault is shared proportionately to the users deposit amount relative to the entire supplied amount, and can be represented as such:

$$User\ Vault\ Rewards\ =\ Total\ Current\ \ Vault\ Rewards\ \times \frac{User\ Deposit}{Total\ Supplied}$$

In the Pool Facet however there is a *accRewardPerShare* index which will keep track of the conversion rate of rewards per *pToken* asset. The *accRewardPerShare* is updated whenever a deposit, withdraw, or claim vault rewards action is taken. When a user claims their vault rewards the Pool Facet will then transfer the rewards based on the following calculation:

$$User\ Vault\ Rewards\ =\ accRewardPerShare\ \times\ User\ pToken\ Balance$$

To ensure attackers are unable to double claim vault yield, the Pinjam Protocol will have to keep an accounting of the *accumulated rewards* earned. Based on that, the protocol will only give out new rewards earned from the vault and an attacker will not be able to double claim earned yield or take yield that belongs to other depositors, which is represented as such:

$$Token\ Total\ Received\ =\ Total\ Current\ Vault\ Rewards\ +\ Rewards\ Claimed$$
$$New\ Rewards\ =\ Token\ Total\ Received\ \ -\ \ Accumulated\ Rewards$$

The *new rewards* are what will update the conversion rate represented by the *accRewardPerShare* index.

2.6 Governance

The Pinjam Protocol has been designed with community control and participation in mind.

Pinjam Governance features will include:

1. Adding new assets to the protocol
2. Adjusting Interest Rate strategies/targets on all assets
3. Voting on new protocol features & improvements
4. Voting on new Vault Strategies and controlling the flow of unborrowed liquidity

3. Value Accrual

Another priority of the Pinjam Protocol is to accrue as much value as possible to token holders. The Pinjam protocol is intending to give out 100% of protocol revenue to its token holders once the token is live, therefore the incentive of the team is aligned with the community and will attempt to accrue as much value as possible for token holders as the team will be some of the larger holders of the token.

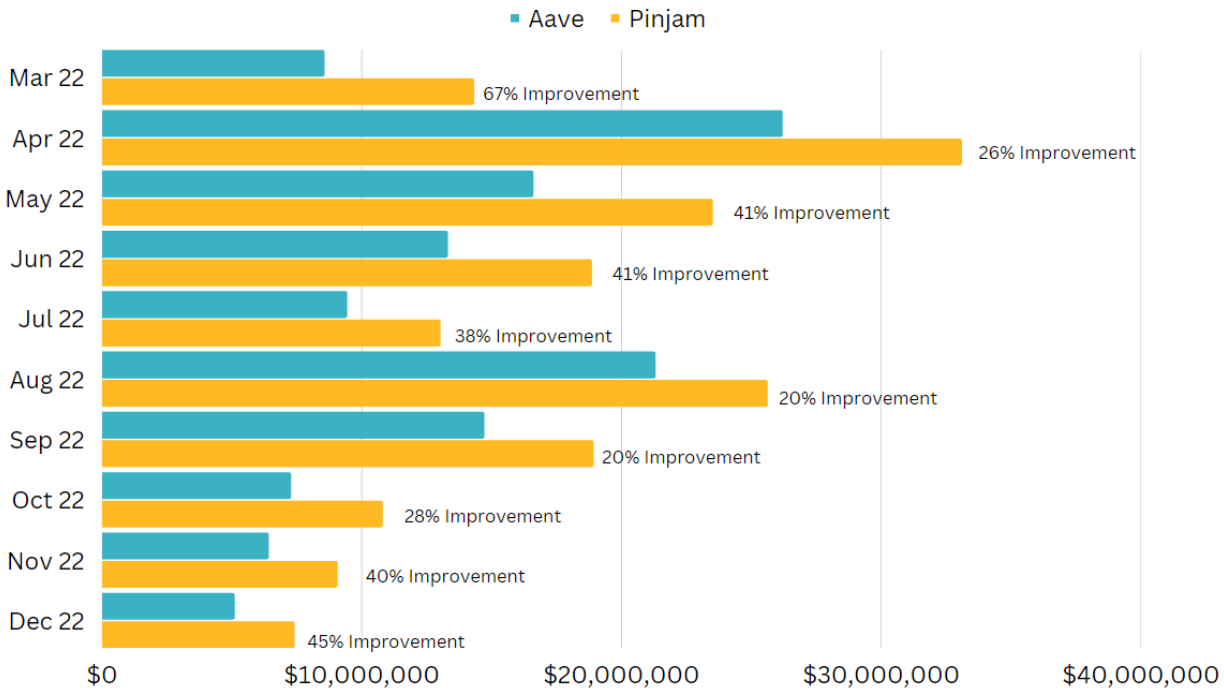
The reason for this is to create enough demand for the token such that there is value when conducting liquidity mining programmes and that it remains a stable source of yield to lenders & borrowers without depreciating too much.

There are currently three primary sources of revenue for the Protocol:

1. Borrowing Fees
2. Farming Fees
3. Liquidation Bonus

Currently the Aave protocol is generating approximately $160,000 a day in revenue on borrowing fees alone with about $5B in TVL with only 40% of assets being borrowed. [10]

Given the same context here's how much more value the Pinjam Protocol will accrue for token holders in the same context as Aave while putting the remaining 60% of assets to work earning just an additional 1% of yield on that Idle Liquidity.



Aave x Pinjam Value Accrual To Token Holders

This shows that the Pinjam DAO would have accrued 20% more value at minimum or 36% more on average, based on a conservative 1% yield earned from just putting 60% of Idle Liquidity to work.

Pinjam intends to be the leading value accruing money market, which will in turn fuel the liquidity mining programme incentivizing further depositors and borrowers activity.

4. Risks of Fractional Reserve Lending

This section covers the traditional & on-chain risk of fractional reserve lending and how the Pinjam Protocol works to minimize risk.

4.1 Traditional Finance

All traditional banks today are fractional reserve banks. The result of this is that most banks are able to achieve higher capital productivity and this results in banks being able to pay depositors to bank with them by giving them a higher depositor interest rate.

The problem with fractional reserve in traditional finance, when you deposit your money into a bank, you do not have any transparency as to what kind of risk the bank is taking. As a matter of fact, when a bank loans money to a business or another bank, they too, will not be fully aware nor have the understanding of the counterparty risk it is taking on.

This is the problem of double entry bookkeeping and is known as hidden counterparty-risk.

This is what makes collapses like FTX possible because investors and lenders do not know how user deposits were being kept or how they were used. [6]

In traditional finance people have to trust the centralized entity that it is not committing accounting fraud in the worst case like Bernie Madoff or in a good case being unknowingly over-levered like the Lehman Brothers, in the 2008 Financial Crisis. [7] [8] [9]

4.2 Decentralized Finance

Being on a blockchain enables triple entry accounting and removes any hidden counterparty-risk due to the open nature of being on-chain.

By being on-chain users can have full knowledge of the counterparty risk the Pinjam Protocol is taking on — such as smart contract risk of Aave or Curve.

This also benefits the Pinjam Protocol by being able to fully evaluate the risk from lending money to a certain protocol and to either permissionless-ly rebalance or withdraw assets from that underlying protocol.

To sum it up, users will be able to enjoy the upside of increased capital productivity while reducing the downside of hidden counterparty risk.

4.3 Economic Risks

By putting idle liquidity to work on other protocols, this opens an attack vector for blackhats to attack the core lending & borrowing infrastructure via market manipulation, potentially leaving the protocol with bad debt.

4.3.1 Lending Strategy Risks

Lending Strategies is defined by lending idle liquidity to base lending protocols such as Aave, Compound, BenQi or Venus. There can be an attack on the Pinjam protocol by temporarily locking access to the funds owned by the Pinjam Protocol.

An example would be during times of high price volatility and users should be liquidated, the Pinjam Vault would have assets deposited into base lending protocols like BenQi. Taking the USDT asset on BenQi for example, an attacker could borrow all the USDT available, and if the Pinjam Vault tries to withdraw USDT, the transaction would revert due to insufficient USDT, which could cause liquidations to fail thus leaving the Pinjam Protocol exposed to bad debt.

To prevent such events from occurring, in a scenario like this the Pinjam Lending Strategy will instead borrow another asset, say USDC and sell it for USDT, providing the assets required to the Pinjam Protocol.

This is an effective method from getting locked out of accessing the funds because for an attacker to borrow, the attacker will need to be overcollateralized depositing another asset to conduct the attack.

Also, an attacker needs to be highly motivated to want to conduct such an attack as it requires extensive capital with no real opportunity of profit other than trying to bring down the Pinjam Protocol. Because if the attacker is borrowing 100% of an entire asset the borrow rate incurred by the attacker will instead profit the deposited asset by the Pinjam Protocol and also cover the borrow of another asset that has a lower borrow rate.

4.3.2 AMM Strategy Risks

AMM Strategies are the type of strategies where idle liquidity is routed through AAMs such as Curve or Uniswap. Due to the complexity of assessing and minimizing AMM risk, the first version of the Pinjam Protocol will not include AMM strategies, but this section will cover a bit of the risk involved and what it might look like to solve those risks.

The biggest risk of routing idle liquidity to an AMM is Impermanent Loss (IL). Heavy impermanent loss could result in the Pinjam Protocol losing funds or receiving back less than what was initially deposited.

This risk can be mitigated by choosing a high liquidity asset pool. To manipulate an ETH/USDT pool for example would be highly expensive and might require tens of millions of dollars, and even if such manipulation worked, it would only be temporary as market forces and arbitrageurs would come in to stabilize the price again.

However, natural market forces could still cause significant Impermenant Loss. This risk can be mitigated by choosing a highly correlated asset pool such as the 3CRV pool of USDT/USDC/DAI. Market manipulation can still occur by depositing significantly more of one asset causing an imbalance. This can once again be prevented by withdrawing the asset that was significantly deposited and selling whatever is required on the open market to acquire the other asset.

5. Conclusion

The Pinjam Protocol improves Decentralized Finance's current offering of decentralized money markets via the increased productivity of Idle Liquidity within the protocol. Resulting in higher value accrual to the protocol and its token holders, offering higher yields to lenders, reducing the cost of borrowing for borrowers and benefiting the entire DeFi ecosystem overall from the Pinjam Protocol acting as a liquidity aggregator.

References

[1] - DefiLlama (2020) Yearn Finance. DefiLlama. Available at: https://defillama.com/protocol/yearn-finance (Accessed: January 29, 2023).

[2] - Aave (2020) Aave Protocol Whitepaper V1. Aave. Available at: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf (Accessed: January 28, 2023).

[3] - Aave (2021) Aave Protocol Whitepaper V2. Aave. Available at: https://github.com/aave/protocol-v2/blob/master/aave-v2-whitepaper.pdf (Accessed: January 28, 2023).

[4] - (2020) Venus Whitepaper. Swipe Wallet. Available at: https://venus.io/Whitepaper.pdf (Accessed: January 28, 2023).

[5] - Aave Deposits & Outstanding Loans Timeseries (no date) Dune Analytics. datanut. Available at: https://dune.com/queries/37478/74298 (Accessed: January 28, 2023).

[6] - Reiff, N. (2023) The collapse of FTX: What went wrong with the Crypto Exchange?, Investopedia. Investopedia. Available at: https://www.investopedia.com/what-went-wrong-with-ftx-6828447 (Accessed: January 31, 2023).

[7] - Singh, M. (2022) The 2007-2008 financial crisis in Review, Investopedia. Investopedia. Available at: https://www.investopedia.com/articles/economics/09/financial-crisis-review.asp (Accessed: January 31, 2023).

[8] - Hayes, A. (2023) Bernie Madoff: Who he was, how his ponzi scheme worked, Investopedia. Investopedia. Available at: https://www.investopedia.com/terms/b/bernard-madoff.asp (Accessed: January 31, 2023).

[9] - Lioudis, N. (2023) The collapse of Lehman Brothers: A case study, Investopedia. Investopedia. Available at: https://www.investopedia.com/articles/economics/09/lehman-brothers-collapse.asp (Accessed: January 31, 2023).

[10] - DefiLlama (2021) Aave Fees. DefiLlama. Available at: https://defillama.com/fees/aave (Accessed: January 29, 2023).