

On November 10th, we put forth our proposal for [a revised Aave token architecture](https://www.delphidigital.io/reports/aaave-token-architecture-v2-towards-aaave-as-an-insured-credit-protocol/). We received incredible feedback from the Aave community and V2 seeks to incorporate much of it. To read the full proposal, see the link on our website here: <https://www.delphidigital.io/reports/aaave-token-architecture-v2-towards-aaave-as-an-insured-credit-protocol/>

Introduction

Before moving onto our suggested design, it's important to understand how the current Safety Module works as well as its flaws.

Aave's Safety Module is effectively an insurance product, albeit a simple, unlevered one. The current Safety Module insurance model consists of a single pool which underwrites Aave protocol risk, including smart contract risk, oracle failure risk, and liquidation risk. Assuming the recent [ARC](#) is passed the Safety Module will be composed of AAVE and Balancer Pool Tokens (BPT) of an 80/20 AAVE/ETH Balancer Pool. In case a shortfall event is triggered by governance, up to 30% of the Safety Module is auctioned off to make the protocol whole. If this 30% isn't enough, the Aave issuance model will issue Aave and auction it off until the insolvent market is made solvent.

We believe the current design has a few key flaws:

- Capital Inefficient:
 - Insurance relies on leverage for efficiency, meaning each \$1 in the capital pool must be used to underwrite >\$1 of risk. This in turn can only be achieved by underwriting a diversified set of risks such that not all claims are due simultaneously. Unlevered insurance products will result in low returns for underwriters and high prices for users.
 - In addition, the unified capital pool bundles different risks together and offers a blended return that will appeal to a narrower capital base.
- Insurance relies on leverage for efficiency, meaning each \$1 in the capital pool must be used to underwrite >\$1 of risk. This in turn can only be achieved by underwriting a diversified set of risks such that not all claims are due simultaneously. Unlevered insurance products will result in low returns for underwriters and high prices for users.
- In addition, the unified capital pool bundles different risks together and offers a blended return that will appeal to a narrower capital base.
- Systemic Risk:
 - The unified capital pool means that any new market added to Aave is automatically insured, introducing unlimited contagion and systemic risk.
 - This in turn will hamper the speed at which Aave can innovate by increasing the costs of potential failed experiments.
 - The unified capital pool means that any new market added to Aave is automatically insured, introducing unlimited contagion and systemic risk.
 - This in turn will hamper the speed at which Aave can innovate by increasing the costs of potential failed experiments.
- Cover demand bundled in: With cover demand being bundled in with the money market product, it's impossible to determine how much underwriters should be paid as well as what cover capacity actually is. This is unlikely to give users certainty that they'd be paid out in case of a Shortfall Event.
- Pro-cyclical: With the Safety Module being denominated in \$AAVE and insurance liabilities being denominated in other assets (mostly USD), there is likely to be significant correlation and reflexivity between Shortfall Events and declines in the value of \$AAVE, especially considering the potential of a mint.

Note: While pro-cyclicality is a real issue, it is more related to the way the capital pool is managed vs the insurance design per se. Our proposals will thus focus primarily on addressing the first three issues.

In our initial suggested solution, we had self-sovereign, independent aDAOs which each operated and governed their own money-markets, earning the majority of the fees generated. We chose this design because, in the existing design, insurance is bundled in with the money markets such that demand for insurance is impossible to compute separately from money market demand. Without a market-based mechanism to determine cover price and capacity, our design instead offloaded these decisions to independent aDAOs who bore both the risk and reward relating to the money markets they managed. This, we hoped, would incentivize good decisions by the principle of skin in the game, ensuring those making decisions are forced to bear the potential downside if they go wrong.

In our new design, insurance is offered as a separate product on the demand side. This makes it possible to compute cover demand and capacity precisely and thus scale back some of the independence provided to aDAOs.

Based on this insight, we propose two models. The first is a simplified, unlevered version which could act as an evolutionary step towards the second model which we see as the optimal, final form design. In both cases, the demand-side experience is identical: users get access to one-click insurance with a money market style architecture leveraging Aave's current interest rate model to calculate cover pricing based on utilisation rate. The designs differ in terms of their supply-side experience; In the first case, users can only stake to the Safety Module, underwriting all risks and receiving a blended return based on cover demand. In the second case, users also stake to the Safety Module but are then able to boost these returns by staking to specific contracts, taking on different amounts of risk and being rewarded accordingly.

Unified Insurance Pool Model

We will begin by describing the UX on both the demand and supply-side. We will then go through an example model with some numbers to illustrate our point.

Demand-side UX

A simple UX where the user can gain one-click insurance. Rather than paying for it upfront as with current decentralised insurance designs, the user pays for it on an ongoing basis as the cover rate is subtracted from his deposit APY. The UX steps are as follows:

- Aave front-end displays two deposit APY interest rates for each money market: the current, uninsured interest rate and a lower one with insurance.
- Similar to the current Aave interest rate model, there could also be stable and fixed interest rates with the latter providing certainty similar to that of buying cover in existing solutions.
- Similar to the current Aave interest rate model, there could also be stable and fixed interest rates with the latter providing certainty similar to that of buying cover in existing solutions.
- If the user selects the insured deposit, he can do so in one click, with the following two transactions happening in the background:
 - User deposits \$USDC, receives \$aaUSDC (insured \$USDC).
 - User's \$aaUSDC would accrue value at the insured APY rate (money market APY - insurance cost).
- Interest rate paid is determined by the Safety Module's utilisation rate just as any other money market pool on Aave (see supply-side UX section for more details).
- The insurance cost (paid in \$USDC) would be used to buy \$AAVE, which would then be deposited into the Safety Module as an insurance premium for stakers.
- User deposits \$USDC, receives \$aaUSDC (insured \$USDC).
- User's \$aaUSDC would accrue value at the insured APY rate (money market APY - insurance cost).
- Interest rate paid is determined by the Safety Module's utilisation rate just as any other money market pool on Aave (see supply-side UX section for more details).
- The insurance cost (paid in \$USDC) would be used to buy \$AAVE, which would then be deposited into the Safety Module as an insurance premium for stakers.
- The user's deposit APY interest is automatically deposited as \$aUSDC and his cost of insurance is automatically deducted. The insured user earns the net of interest from \$aUSDC deposit APY and the cost of insurance.
- If the user wants to withdraw his \$USDC, he can also do so in one-click, with the following happening in the background:
 - User withdraws his \$aaUSDC to \$USDC.
 - As \$aaUSDC supply decreases, capacity opens up in the Safety Module.
- Extra: It would make sense to have automated features such that users can automatically pay back their insurance if the net yield goes negative or below a certain threshold.
- User withdraws his \$aaUSDC to \$USDC.
- As \$aaUSDC supply decreases, capacity opens up in the Safety Module.
- Extra: It would make sense to have automated features such that users can automatically pay back their insurance if the net yield goes negative or below a certain threshold.

Supply-side UX

The supply-side would work similarly to the current experience:

- Users would be able to lock their \$AAVE into the Safety Module and receive \$stkAAVE at a 1:1 value ratio.
- Users would see an Insurance Yield based on the current utilization of the Safety Module.
- Governance would determine the maximum cover capacity per pool and the global minimum capital requirement (MCR) in relation to the size of the Safety Module.

[
1003×925
](https://lh5.googleusercontent.com/4JT6-8LQOU2Qzizi1oSc_AxkfYwaQAFzCgEGbHZuFWy6pS9NSwQV2c8XcZelRN4jajNP6TzMCqQfngxEYCI61wPvzp4_6LBmYLdJo4TgLfdgzgntyGfGMLrOjwXUIJDTjcmcyzK)

Example Model

To exemplify the above, let’s take a closer look at what this would look like for a user looking to insure their USDC deposit.

First, let’s consider the demand side. The two most important variables for a user looking to insure their USDC deposits are: 1) the USDC money market APY and 2) the insurance cost. The APY would work in the same way as it works in the current Aave model, which depends on the interest rate model and utilization. In a similar fashion, the insurance cost, which is effectively an interest rate, would depend on the interest rate model used. Given that Aave already has an interest rate model, we propose the same model for the insurance pools, albeit with slightly different parameters.

Compared to a money market pool, we see the insurance pool interest rate model working in the following way:

[
960×282
](https://lh4.googleusercontent.com/qHRx30nvcWMhC5LKemnWS4KE2l8yo3ixW-Zai0LEXXnERswNeUhmdeEn_1pGORX60Kx-Jex99-6K28rbIoAYF_9l4uGnGN3f4n05Vk3D9iY3kzCHlwmPSDJJCep72earlTWbDuXm)

The previous parameters, which would be chosen by governance, are what determine the insurance pool interest rate model. Compared to the money market parameters, we propose working with a lower R0 to account for the fact that money market APY depends on utilization, which starts at 0.

In the following figure, we show what this interest rate model looks like at the same utilisation levels (money market and insurance pool). The difference between the APY (blue line) and the insurance cost (orange line) is the insured APY (grey line). Notice that at the same utilisation levels, there’s a minimum utilisation threshold from where it makes sense to buy insurance (in this case around 20%), since any lower utilisation level would imply a negative insured APY for the user. Nevertheless, it is possible some users may choose to incur negative insured APY in order to access leverage while being sure their deposits are protected.

[
1341×731
](https://lh3.googleusercontent.com/cLhGz0Owhjtp-mTi3gH3orx1nb9-2_BrsO44RfP-0egborR_n02lvjqko1u0VECX9t7SQXv1n2xiKmuZ42a9sP50VMKwz8lkEVq2rd1vPLQZJWPsxHhJc09XoF9AheHmIXHlXnBB)

Note: For illustration purposes, the chart only goes to 75% utilisation. After 75%, interest rates rise sharply which makes visualizing the point more difficult

In practice, it’s unlikely that the utilisation rate for the money market and the insurance pool will be the same at any given point in time. The following table shows what the insured APY (money market APY – insurance cost) looks like at different utilization levels. As was hinted before, it clearly illustrates how the insured APY depends on the utilization levels of the money market and the insurance pool. Cells highlighted in green indicate positive insured yields, while non-highlighted cells indicate the opposite.

[
1162×620
](https://lh4.googleusercontent.com/psVQy0a1-5eWcmMYMNL3lXhXYo6fXJpNjXqi4IPA4vf0MQsxqzOx_o3nm21dNyhyhfBwaXflbp8f-Xrx7cXund7y5UAWU_dT4Uei_to00jz9ON8ZNzZ6-RYdUXo2Kq6UcsYkgse)

On the supply side the insurance yield would depend on the utilisation (by all money markets and insured products) of the cover capacity of the Safety Module:

[
364×552
](https://lh6.googleusercontent.com/P93yHjFy1HvGXZHOhy2t8a1vJl9ImHjofSbccV-Pb1LKGeorleb9Vvxnu2Qs6xEv3tsGrAQKZQevHRcXjV15QdhdRsbEn67iqCciJAEyWXvTfAfq1y55NOUkPqoi5CQbgSpaEmc)

Lastly, the effective yield for Safety Module stakers would depend on the minimum global collateralization factor (that determines global capacity) and the share of the fees that get distributed to the Safety Module (versus the Aave Reserve).

Discussion

We believe this model mitigates a few of the flaws with the Safety Module that we highlighted in the introduction.

Firstly, contagion and systemic risk are mitigated since users must buy cover rather than have it bundled in, meaning the Safety Module’s liabilities are constrained and can be computed precisely. As a result, permissionless innovation can happen quicker since decisions about listing assets are separate from decisions about insuring those assets. Aave could theoretically list an asset that it chooses not to insure with the Safety Module or that it chooses to insure only up to a certain limit.

The primary drawback of this model is that it imposes a uniform insurance cost across all Aave money markets which doesn’t account for the differing risks posed by the different assets being underwritten. In a rational market, an equilibrium price may be reached at an insurance cost higher than what users are willing to pay for less risky assets but lower than what users are willing to pay for risky assets. This will lead to adverse selection in which riskier insurance demand crowds out demand for less risky assets. While this could be solved via governance defining the maximum cover capacity per pool, this is inelegant as it defers to governance what could be determined via market-based mechanisms.

In addition, while more efficient than the current Safety Module, in its current iteration this model is still unlevered. While leverage could be added and determined at the governance level, the aforementioned flaws of the uniform insurance cost are amplified by leverage since users can now take out even more cover on risky assets. This could be mitigated by governance determining weights but we believe it’s better to use market-based mechanisms where possible.

Segregated Insurance Pools

To solve the issues with the Unified Insurance Pool Model, we propose Segregated Insurance Pools similar in spirit to our original aDAO proposal.

On the demand-side the UX works similarly to the Unified Insurance Pool Model as users still have access to one-click insurance. However, rather than paying a single insurance cost across the entire protocol, each money market pool would have its own insurance cost. This is enabled by a more complex supply-side UX in which \$stkAAVE holders can choose to stake into individual pools, earning a higher share of that pool’s fees in exchange for acting as the first tranche of risk capital in case of a Shortfall Event.

We believe this model enables the greatest capital efficiency by using market-based mechanisms on both the demand and supply side to determine cover pricing and returns to stakers. Leverage can still be determined at the protocol level but capacity will be allocated at the insurance pool level.

Demand-side UX

The demand-side would work in a similar way to the previous implementation:

- Let’s assume the user is depositing into the USDC pool. If the user selects the insured deposit, he can do so in one click with the following happening in the background:
- User deposits \$USDC, receives \$aaUSDC (insured \$USDC).
- User’s \$aaUSDC would accrue value at the insured APY rate (money market APY - insurance cost). Insurance cost would be determined by capacity in the USDC insurance pool.
- The insurance cost (paid in \$USDC) would be used to buy \$AAVE, which would then be deposited into the \$aUSDC Insurance pool, the \$stkAAVE pool and the AAVE Reserve at a ratio determined by the community.
- User deposits \$USDC, receives \$aaUSDC (insured \$USDC).

- User's \$aaUSDC would accrue value at the insured APY rate (money market APY - insurance cost). Insurance cost would be determined by capacity in the USDC insurance pool.
- The insurance cost (paid in \$USDC) would be used to buy \$AAVE, which would then be deposited into the \$aUSDC Insurance pool, the \$stkAAVE pool and the AAVE Reserve at a ratio determined by the community.
- If the user wants to withdraw his \$USDC, he can also do so in one-click, with the following happening in the background:
- User withdraws his \$aaUSDC to \$USDC.
- As \$aaUSDC supply decreases, capacity opens up in the USDC insurance pool.
- User withdraws his \$aaUSDC to \$USDC.
- As \$aaUSDC supply decreases, capacity opens up in the USDC insurance pool.

Supply-side UX

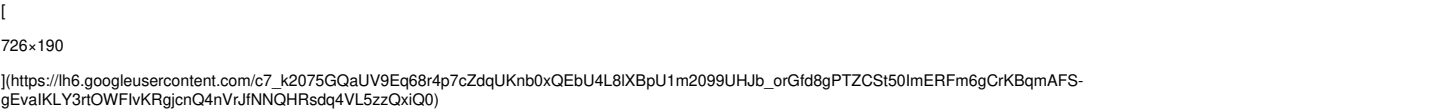
On the supply-side, users can take different amounts of risk based on their preference. The UX steps can be seen as follows:

- Users can stake to the Safety Module by converting \$AAVE to \$stkAAVE at a 1:1 value ratio. That means that a user who deposits 10USD worth of \$AAVE would get 10USD worth of \$stkAAVE back. As fees are deposited into the \$stkAAVE Pool, it is possible that the value of \$stkAAVE (which represents a share of the \$stkAAVE Pool) differs from the value of \$AAVE.
- Users can take their \$stkAAVE and lock it into specific contracts they want to insure with some amount of leverage as determined by governance.
- For instance, if the leverage is set at 10x, each unit of \$stkAAVE can be staked into 10 different insurance pools.
- For instance, if the leverage is set at 10x, each unit of \$stkAAVE can be staked into 10 different insurance pools.
- If a user wants to stake to the \$aUSDC contract, the following happens:
- User locks \$stkAAVE and receives \$aUSDCpool tokens issued on a virtual bonding curve.
- The fees generated from the \$aUSDC Insurance Pool would be shared between the \$aUSDC Insurance Pool, the \$stkAAVE Pool and the AAVE Reserve at a ratio determined by the community.
- The amount of \$stkAAVE determines overall insurance cover capacity and maximum capacity per pool, whereas the amount of \$stkAAVE in the \$aUSDC pool determines capacity for \$aUSDC cover, effectively shifting the mechanisms to a per pool, market based capacity computation.
- Similar to the aDAO bonding curves, the bonding curve is locked at the minimum capital requirement required to satisfy active cover (MCR) at whatever the leverage ratio is. Users will not be able to withdraw while the bonding curve is locked.
- User locks \$stkAAVE and receives \$aUSDCpool tokens issued on a virtual bonding curve.
- The fees generated from the \$aUSDC Insurance Pool would be shared between the \$aUSDC Insurance Pool, the \$stkAAVE Pool and the AAVE Reserve at a ratio determined by the community.
- The amount of \$stkAAVE determines overall insurance cover capacity and maximum capacity per pool, whereas the amount of \$stkAAVE in the \$aUSDC pool determines capacity for \$aUSDC cover, effectively shifting the mechanisms to a per pool, market based capacity computation.
- Similar to the aDAO bonding curves, the bonding curve is locked at the minimum capital requirement required to satisfy active cover (MCR) at whatever the leverage ratio is. Users will not be able to withdraw while the bonding curve is locked.
- If a Shortfall Event occurs, \$stkAAVE holders determine whether or not that event is covered by the insurance policy. Shortfall Events in particular contracts will wipe out stakers to those pools first by burning some of their \$stkAAVE
- As soon as a Shortfall Event is triggered, stakers to contracts are frozen as to restrict them from exiting before a major insurance event is concluded.
- As soon as a Shortfall Event is triggered, stakers to contracts are frozen as to restrict them from exiting before a major insurance event is concluded.
- To simplify the UX, contracts can be grouped into indexes based on their risk characteristics such as \$Apool, \$AAPool, \$BBBpool, etc. If a user stakes his \$stkAAVE into one of these indexes, he is automatically staking into all underlying pools in proportion to their size.
- Anyone can permissionlessly add a money-market but it is only listed on the Aave front-end if \$stkAAVE holders vote to insure it and it attracts a minimum amount of \$stkAAVE.

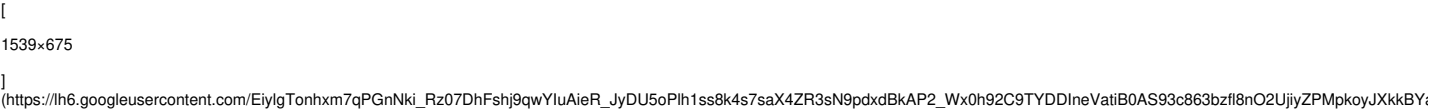


Example Model

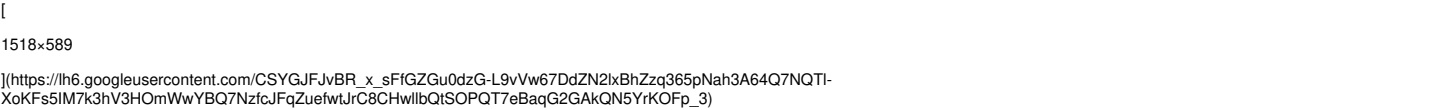
The main difference with this model is that it allows for each insurance pool to have its own interest rate model and its own capacity. As such, important risk parameters could be chosen for each pool specifically taking into account their own particularities. Let's explore an example for a USDC and a YFI pool. Since USDC is less risky, we could set a higher optimal insurance utilisation level for this pool. The next tables show what the interest rate models could look like for these two assets.



As can be seen, the YFI insurance pool has a lower optimal utilisation level and a higher Rslope2, which accounts for its higher risk. This would, in turn, make insurance above the 45% utilisation level highly expensive. The next graph depicts the insurance cost for this market at different utilisation levels. What is clear from it is that, at utilisation levels above 45% insurance starts getting highly expensive, which would incentivise users to decrease use of the insurance pool.



Let's compare this to what the USDC graph looks like. What can be seen with the following graph is that insurance cost only starts to spike to highly expensive levels when utilisation hits 90%, the optimal utilisation level for the USDC insurance pool.



What the above graphs show is that the flexibility to choose the insurance interest rate model for each market has clear advantages as they can be adjusted for the riskiness of each market.

In addition, this model allows for the size of each insurance pool to be determined by a market-driven approach, as it depends on the amount of \$stkAAVE staked to each particular pool and the effective leverage of the protocol. If, for example, the maximum leverage allowed by the protocol is 10x, each user gets to stake his \$stkAAVE into 10 different insurance pools. To illustrate this point, let's assume a user holds 100 \$stkAAVE. If he chooses to only stake into one pool, then that pool's capacity would be increased by 100 \$stkAAVE and would be effectively backed by 100 \$stkAAVE. On the other hand, that user may use leverage and stake 100 \$stkAAVE into two pools (or pool indexes), 100 \$stkAAVE each. In this case, each pool's capacity would increase by 100 \$stkAAVE, but they would only be backed by 50 \$stkAAVE, meaning that they're leveraged up to 2x (assuming no other stakers for simplicity).

In this sense, this model not only allows for a tailored approach to pricing dependent on the risk characteristics of each market, but also to sizing, as the capacity of each insurance pool would depend on the amount of \$stkAAVE staked to it.

Lastly, with this model, an individual staker's (\$aUSDCpool holder) effective yield would depend on his leverage and the individual yields of the pools he's insuring. In a similar fashion, the stkAAVE

pool's yield would depend on all the individual pools' yields and the share of those fees that gets distributed to the main stkAAVE pool.

Conclusion

We really appreciate all the excellent feedback and discussion following our first proposal. We've incorporated many of the learnings from those discussions into V2. We've also sought to provide an achievable path towards implementation. The simpler unified insurance pool design would act as a first step which would also serve to get demand-side users used to buying insurance. The more complex segregated insurance pool design can be implemented later, providing greater capital efficiency and granularity in cover pricing.

It's worth noting that, in addition to serving as an insurance product to be used by the current Aave money market users, this architecture allows for Aave to offer a generalized insurance product that could extend beyond its current proposed use case. For instance, this insurance architecture could be used to insure protocols that integrate with the Aave platform, such as Aavegotchi. Going even further, this product could be spun off as its own independent insurance platform (controlled and governed by AAVE holders) that would offer a wide range of insurance covers for any accepted projects. To prevent insurance demand from other projects crowding out Aave money market insurance demand, a minimum share of the Safety Module could be exclusively dedicated to cover Aave, with the remaining being distributed among other protocols.

What's even more important to highlight here, beyond any particular details of what a full-blown insurance product might look like, is that the Safety Module is a critical and flexible piece of infrastructure of the Aave protocol. Thus, with the above consideration we look to encourage discussion around different strategies to effectively use this potent piece of infrastructure going forward. We believe the Safety Module is a competitive advantage for Aave and its effective use to be a key consideration for the future of the protocol.

We look forward to the community's feedback on our proposals and will be available to answer any questions.

[Jose Maria Macedo](#) and [Jonathan Erlich](#)