

The mangrove (<https://mangrove.exchange>) team would like to submit the following proposal that should be of use for many integrators managing user funds on AAVE. This follows the discussion on issue #691

([repay should not throw when repaying a borrow of the same block · Issue #691 · aave/aave-v3-core · GitHub](#)) on github.

Summary

The AAVE V3 protocol disallows borrowing and repaying an asset in the same block. This restriction does not exist in AAVE V2 and can harm or break composability for integrators (such as Mangrove), so we simply propose removing this feature.

Specification of the problem

AAVE V3 adds a new check forbidding the borrow

-repay

or the repay

-repay

sequence in a single block. More specifically, the debt of an address for a specified asset cannot be repaid if the associated debt token of the user has been interacted with in the same block.

This restriction might become a blocker for integrators that rely on AAVE pools because their accounts on AAVE integrate multiple user accounts on their side. For example, user A

of protocol P

repaying its debt on AAVE via P

after user B

of the protocol tries to borrow from P

is interpreted as an illegal sequence by the pool's logic that identifies users A and B as P

. An adversary could exploit this sequence to halt proper functioning of integrations on AAVE V3.

Notably, any address can repay assets on behalf of any address, thus opening the door to DOS attacks preventing addresses from repaying their debts.

This restriction does not seem to be mentioned in the documentation.

The example of Mangrove

Mangrove is an exchange that follows an offer-is-code approach. This means that every Mangrove offer comes with the address of a contract which the market engine calls when the offer is matched, at which time the said contract will typically obtain the needed liquidity from a specified source such as AAVE.

The contract often manages multiple offers on a given market. One desirable use case is fulfilling a bundle of asks with volume v_1

at price p_1

, v_2

at p_2

p_1

, etc. If the liquidity manager first borrows v_1

, v_2

, etc. during the execution of an incoming market order, the borrow costs would be paid per matched order. A more desirable sequence would be to borrow an amount likely to cover all ask orders being executed once at the first match, and then to repay the excess liquidity at the end of the transaction while also depositing the amount of cash obtained from the taker.

In so doing, the contract satisfies the property that any execution will have a near constant gas cost regardless of order size. This property is important for the taker experience. However, for this optimization to be viable, it is key that AAVE allows addresses to borrow and repay assets in a single transaction in the same manner as flash loans.

Conclusion

The prohibition of borrow-repay in the same block is a very general problem, and many integrators could face this issue. We noticed that this issue is often asked about in the AAVE Discord.

Moreover, this affects the backward compatibility of AAVE V3 with AAVE V2. This could even be dangerous for integrators if AAVE V2 is upgraded into AAVE V3.

Let us know what you think!