

# System Stabilizer Module

Keeping the Maker Protocol Stable \* Module Name: \* System Stabilizer \* Type/Category: \* DSS —> System Stabilizer Module (Vow.sol, Flap.sol, Flop.sol) \* [Associated MCD System Diagram](#) \* Contract Sources: \* [Vow](#) \* [Flop](#) \* [Flap](#) \* \* \*

## 1. Introduction (Summary)

The System Stabilizer Module's purpose is to correct the system when the value of the collateral backing Dai drops below the liquidation level (determined by governance) when the stability of the system is at risk. The system stabilizer module creates incentives for Auction Keepers (external actors) to step in and drive the system back to a safe state (system balance) by participating in both debt and surplus auctions and, in turn, earn profits by doing so.

For a better understanding of how the Auction Keepers relate to the System Stabilization Mechanism, read the following resources:

- [Auctions & Keepers within MCD 101](#)
- [How to Run Your Own Auction Keeper Bot in MCD Blog Post](#)
- [Auction Keeper Bot Developer Setup Guide](#)
- 

## 1. Module Details

The System Stabilizer Module has 3 core components consisting of theVow ,Flop , andFlap contracts.

### System Stabilizer Components Documentation

- [Vow Documentation](#)
- [Flop Documentation](#)
- [Flap Documentation](#)
- 

## 1. Key Mechanisms & Concepts

### Summary of theSystem Stabilizer Module Components

1. TheVow
2. represents the overall Maker Protocol'sbalance
3. (both system surplus and system debt). The purpose of thevow
4. is to cover deficits via debt auctions and discharge surpluses via surplus auctions.
5. TheFlopper
6. (Debt Auction) is used to get rid of theVow
7. 's debt by auctioning off MKR for a fixed amount of internal system Dai. Whenflop
8. auctions are kicked off, bidders compete with decreasing bids of MKR. After the auction settlement, the Flopper sends received internal Dai to theVow
9. in order to cancel out its debt. Lastly, the Flopper mints the MKR for the winning bidder.
10. TheFlapper
11. (Surplus Auction) is used to get rid of theVow
12. 's surplus by auctioning off a fixed amount of internal Dai for MKR. Whenflap
13. auctions are kicked off, bidders compete with increasing amounts of MKR. After auction settlement, the Flapper burns the winning MKR bid and sends internal DAI to the winning bidder.
- 14.

### How the System Stabilizer Module Interacts with the other DSS Modules

?

### How do theVow ,Flop andFlap contracts help the MCD system operate?

?

### Vow

When the Maker Protocol receives system debt and system surplus through collateral auctions and CDP stability fee accumulation, it will deviate from system equilibrium. The job of the Vow is to bring it back to system equilibrium.

### The Priority of the Vow

1. To kick-off debt and surplus auctions (Flop and Flap), which in turn, corrects the system's imbalances.
- 2.

## Flop

The purpose of debt auctions is to cover the system deficit, which is resembled by  $Sin$  (the total debt in the queue). It sells an amount of minted MKR and purchases Dai that will be canceled 1-to-1 with  $Sin$ .

The Priorities of the Flopper:

1. To raise an amount of Dai equivalent to the amount of bad debt as fast as possible.
2. To minimize the amount of MKR inflation.
- 3.

?

## Flap

The purpose of the surplus auctions is to release Dai surplus from the Vow while users bid with MKR, which will be burned, thus reducing the overall MKR supply. It will sell a fixed amount of Dai to purchase and burn MKR.

The Priority of the Flapper:

1. To mechanically reduce the MKR supply when auctioning off Dai surplus.
- 2.

?

1. Gotchas (Potential Sources of User Error)

## AuctionKeepers

In the context of running an Auction Keeper to perform bids within an auction, a primary failure mode would occur when a Keeper specifies an unprofitable price for MKR (more info [here](#)).

- This failure mode is due to the fact that there is nothing the system can do to stop a user from paying significantly more than the fair market value for the token in an auction (this goes for all auction types, flip
- , flop
- , and flap
- ).
- Keepers that are performing badly are primarily at risk during the debt
- phase since they could return too much collateral to the original CDP and end up overpaying (i.e. pay too much Dai
- (bid
- ) for too few gems (lot
- )).
- 
- - Note:
- - This does not apply to the Flap contract (Flop and Flip only)
- \*
- 

When no Auction Keepers Bid:

- For both the Flip
- and Flap
- auctions, the tick
- function will restart an auction if there have been 0 bids and the original end
- has passed.
- In the case of a Flop
- auction expiring without receiving any bids, anyone can restart the auction by calling tick
- . Along with restarting the auction, it also includes a necessary increase of the initial lot
- size by pad
- (default to 50%). This extra process is because the lack of bidding for the lot
- could be due market circumstances, where the lot
- value is too low and is no longer enough for recovering the bid
- amount.
- 

1. Failure Modes (Bounds on Operating Conditions & External Risk Factors)

## Governance Trade-offs

- beg
  - - If too high, it would discourage bidding and create a less efficient auction mechanism.
  - - If too low, it would not be that significant but would encourage small bids and increase the likelihood of theBid.end
  - - being hit before the "true" price was found.
- \*
- ttl
  - - If too high, it would cause the winning bidder to wait "too long" to collect their winnings (depending on the situation, possibly subjecting them to market volatility).
  - - If too low, it would increase the likelihood of bids expiring before other bidders get a chance to respond.
- \*
- tau
  - - If too high, it would not have that significant of an impact as auctions would still operate normally based on theBid.tic
  - - .
  - - If too low, it would increase the chance that an auction will end before the "true" price was found.
- \*
- 

Other Non-Obvious Auction Failure Modes (This is for all auction types: Flip, Flop, Flap)

#### Potential Issues around Network Congestion

When auctioning off collateral tokens, bids are finalized after an interval of time (ttl ) has passed. Hence, in the case when extreme network congestion occurs,ttl and auctions are affected because they can take longer than three hours to confirm a transaction. Therefore, due to Ethereum network congestion, this can result in auctions settling for less than the fair market value. Due to this potential issue, bidders need to calculate network congestion risks when making bids.

Example:

When high network congestion occurs, where common APIs can be shut down due to denial of service (DoS), and where high gas prices are present, this can result in bidding becoming extremely expensive regardless of whether bids win or not. Unfortunately, this also results in only enabling those who have the experience and technical knowledge to build their own transactions are able to use and participate in auctions. Therefore, auctions thus settle for less than the fair market value.

Audit Section Source: Timestamp dependence in auctions enables denial of service([Page 8](#))

#### Transaction-reordering / Front Running Attacks on Auctions

Front running attacks are possible. In order to mitigate this possibility, we reviewed and evaluated other auction options such as [dutch](#) and/or commit reveal but do not currently feel this change is worth delaying the whole system for. Due to the modular nature of the system and the fact that the auction modules can be swapped out, it will be possible for governance to upgrade the auction process in the future.

[Previous Liquidation 2.0 Module](#) [Next Flapper - Detailed Documentation](#) Last updated 4 years ago On this page \* [1. Introduction \(Summary\)](#) \* [2. Module Details](#) \* [3. Key Mechanisms & Concepts](#) \* [4. Gotchas \(Potential Sources of User Error\)](#) \* [5. Failure Modes \(Bounds on Operating Conditions & External Risk Factors\)](#) \* [Governance Trade-offs](#)

[Export as PDF](#)