

# ESM - Detailed Documentation

The ESM is the trigger system for the shutdown of the Maker Protocol \* Contract Name: \* esm.sol \* Type/Category: \* Emergency Shutdown Module \* [Associated MCD System Diagram](#) \* [Contract Source](#) \* [Etherscan](#) \*

## 1. Introduction (Summary)

The Emergency Shutdown Module (ESM) is a contract with the ability to call `End.cage()` to trigger the Shutdown of the Maker Protocol.

?

## 1. Contract Details

ESM (Glossary)

Key Functionalities (as defined in the smart contract)

rely - Grant an address admin powers

deny - Revoke admin powers from an address

file - Allow admin to update thresholdmin and address ofend

cage - Permanently disable the shutdown module

fire - Trigger shutdown by calling `End.cage`

denyProxy - Following the wards rely/deny pattern, calls deny on a given contract

join - Deposit MKR to the shutdown module

burn - Burn any MKR deposited into the shutdown module

Other

gem - MKR Token contract [address]

wards(admin: address) - Whether an address has admin powers [address: uint]

sum(usr: address) - MKR join balance by user [address: uint]

Sum - Total MKR deposited [uint]

min - Minimum MKR amount required for fire and denyProxy [uint]

end - The End contract [address]

live - Whether the contract is live (not caged) [uint]

## 1. Key Mechanisms & Concepts

MKR holders that wish to trigger Shutdown must join MKR into the ESM. When the ESM's internal `Sum` variable is equal to or greater than the minimum threshold (`min`), the ESM's `fire()` and `denyProxy()` methods may be called by anyone. The `fire()` method, in turn, calls `End.cage()`, which starts the Shutdown process.

The ESM is intended to be used in a few potential scenarios:

- To mitigate malicious governance
- To prevent the exploitation of a critical bug (for example one that allows collateral to be stolen)
- 

In the case of a malicious governance attack, the joiners will have no expectation of recovering their funds (as that would require a malicious majority to pass the required vote), and their only option is to set up an alternative fork in which the majority's funds are slashed and their funds are restored.

In other cases, the remaining MKR holders may choose to refund the ESM joiners by minting new tokens.

Note: Governance can disarm the ESM by calling `cage()` (this is distinct from `End.cage()`).

## 1. Gotchas (Potential Source of User Error)

## Unrecoverable of Funds

It is important for users to keep in mind that joining MKR into the ESM is irreversible—they lose it forever, regardless of whether they successfully trigger Shutdown. While it is possible that the remaining MKR holders may vote to mint new tokens for those that lose them triggering the ESM, there is no guarantee of this.

## Game Theory of Funding and Firing the ESM

An entity wishing to trigger the ESM but possessing insufficient MKR to do so independently must proceed with caution. The entity could simply send MKR to the ESM to signal its desire and hope others join in; this, however, is poor strategy. Governance, whether honest or malicious, will see this, and likely move to de-authorize the ESM before the tipping point can be reached. It is clear why malicious governance would do so, but honest governance might act in a similar fashion—e.g. to prevent the system from being shut down by trolls or simply to maintain a constant threshold for ESM activation. (Honest governance, or even deceptive malicious governance, would be expected to then replace the ESM.) If governance succeeds in this, the entity has simply lost MKR without accomplishing anything.

If an entity with insufficient MKR wishes to trigger the ESM, it is better off first coordinating with others either off-chain or ideally via a trustless smart contract.. If a smart contract is used, it would be best if it employed zero-knowledge cryptography and other privacy-preserving techniques (such as transaction relayers) to obscure information such as the current amount of MKR committed and the addresses of those in support.

If an entity thinks others will join in before governance can react (e.g. if the delay for governance actions is very long), it is still possible that directly sending insufficient MKR to the ESM may work, but it carries a high degree of risk. Governance could even collude with miners to prevent calls, etc if they suspect an ESM triggering is being organized and wish to prevent it.

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

#### Authorization Misconfigurations

The ESM itself does not have an isolated failure mode, but if the other parts of the system do not have proper authorization configurations (e.g. the End contract does not authorize the ESM to call `callcage()` ), then the ESM's `fire()` method may be unable to trigger the Shutdown process even if sufficient MKR has been committed to the contract.

[Previous](#) [End - Detailed Documentation](#) [Next](#) [MCD Glossaries](#) Last updated 1 year ago On this page \* [1. Introduction \(Summary\)](#) \* [2. Contract Details](#) \* [ESM \(Glossary\)](#) \* [3. Key Mechanisms & Concepts](#) \* [4. Gotchas \(Potential Source of User Error\)](#) \* [Unrecoverable of Funds](#) \* [Game Theory of Funding and Firing the ESM](#) \* [5. Failure Modes \(Bounds on Operating Conditions & External Risk Factors\)](#) \* [Authorization Misconfigurations](#)

[Export as PDF](#)