

## Was sind Gauges?

- Vergleichbar mit Gauges von Velodrom / Solidity.
- Durch Anlage der LP Tokens der Gauge erhält man STABLE als reward.
- Dieser Mechanismus soll das Locken von STABLE reward-Token attraktiver gestalten, da man durch das Voting auf Gauges zusätzliche Rewards in Form des sogenannten "Boost-Faktors" erhält. Dieser kann bis zu einem Faktor von 2.5 reichen.
- Der Boost-Faktor erhöht sich anteilig der Votes auf dem jeweiligen Gauge nicht wie bei Solidly über alle Gauges hinweg. Somit hat jede Gauge ein eigenes Voting und es muss auf genau diese Gauge gevotet werden um den Boost zu bekommen. (was zu echten Wettbewerb führen sollte selbst in AutoComputer)
- Gauges zahlen, sobald eine neue Epoche beginnt, jeden Block die rewards aus die sie aus der GaugeFactory bekommen haben
- Gauges zahlen standardmäßig 2% (200) "*referralFee*" über 3 Level [6000, 3000, 1000] aus, wenn "*getRewardForOwnerToOtherOwner*" ausgelöst wird. Der User bekommt nur 98% seiner Rewards ausgeschüttet. Dabei gibt es ein Maximum von 10% referral fees.
- Die Funktion "*claimVotingFees*" übergibt die Fees aus den hinzugefügten LP-Tokens in das Bribing der Gauges (wird extern getriggert).
- Immer wenn "*updateReward*" getriggert wird, wird geschaut ob "*nextPoke*" ausgelöst werden soll um die Votingpower wieder anzupassen, falls der Nutzer nur *claims* aber nicht mehr votet.
- Jeder Nutzer kann die Auszahlung von einem Nutzer triggern.

## GaugeFactory

- Die GaugeFactory ist für die Erstellung und Verwaltung neuer Gauges verantwortlich.
- Bei der Erstellung eines Gauges wird eine neue Anfrage in der BribeFactory ausgelöst und eine Bribe erstellt.
- Ermöglicht einzelnen Nutzern mit unseren veTokens (veSTABLE und veUSDFI) über den Voter ([veVoteProxy](#)) die Vergabe der STABLE-rewards inklusive des Boost-Faktors an einzelne Gauges.
- Im Gegenzug müssen die Nutzer mit ihren veToken auf die jeweiligen Gauges voten um diese Gelder zu erhalten.
- Erstellt bei jedem Auslösen von "*preDistribute*" eine neue Epoche "*epoch++*" die für bribes benötigt wird
- Die Funktion "*preDistribute*" kann nur 1x pro Woche ausgeführt werden und wird immer wieder auf "*lastDistribute*" +1 Woche festgelegt. Somit wird ein Verschieben der Fixzeit verhindert, auch wenn es etwas später getriggert wird als nach einer Woche.
- Die GaugeFactory holt automatisch die rewards aus dem StableMinter ab wenn "*preDistribute*" getriggert wird.
- Voting ist nur im Voter möglich und wird von dort aus übergeben. Der Voter kann upgedatet werden, um eventuelle Votingmissbräuche eindämmen zu können.
- Ein Nutzer kann nur einmal pro Epoche voten. Es ist möglich nach dem Voten jedoch seine Stimmen zu resettet, wodurch es aber nicht erneut möglich ist zu voten.
- Neue Gauges können nur von einem Admin oder der Governance erstellt werden um Scam Gauges zu vermeiden. Im späteren Verlauf wird diese Funktion weiter dezentralisiert.
- Gauges haben ein "*maxVotesToken*" (max. 100000 (100%)), um zu verhindern, dass ein Gauge zu viele STABLE-Rewards bekommt, obwohl diese wenig Erträge für das Ökosystem generiert.
- Gauges können den Status "*deprecateGauge*" oder "*resurrectGauge*" erhalten nach der Erstellung von einem Admin / Gov
- Nach "*preDistribute*" kann "*distribute*" ausgelöst werden um die Rewards zu verteilen (für jede Gauge einzeln)
- ein Voter kann egal an welchen Tag nur 1x pro "Epoch" voten

## Bribes

- Das Bribing-System soll ein permanentes voting auf die gleichen Farms verhindern. Hier erhalten die Bereitsteller der Liquidität in den Farms die Möglichkeit, Bestechungsgelder zu verteilen.
- 20% der LP fees die erzeugt werden im Hinterlegten LP in dem dazugehörigen Gauge werden hier jeder epoche hinzugefügt
- Bribes zahlen standardmäßig 2% (200) "referralFee" über 3 Level [6000, 3000, 1000] aus, wenn "getRewardForOwnerToOtherOwner" ausgelöst wird. Der User bekommt nur 98% seiner Rewards ausgeschüttet. Dabei gibt es ein Maximum von 10% referral fees.
- Es können weitere externe Tokens als reward "rewardPerToken" hinzugefügt werden um noch mehr Anreize zu geben
- Ein Owner kann eine andere Adresse Whitlisten "setWhitelisted", die seine rewards abholen kann, um sie an dieselbe wallet zu senden. Die Idee dahinter ist eine wallet zu haben, in der die LPs liegen und eine, auf der die rewards eingehen (hot / cold wallet).
- Jeder Nutzer kann die Auszahlung von einem Nutzer triggern.
- Mit "getRewardForOwnerToOtherOwnerSingleToken" kann ein einzelner reward token abgeholt werden in case ein anderer ist boken / paused
- Bribes folgen der "epoch" der "GaugeFactory"
- Ein Nutzer hat 50 "epoch" zeit (ca 1 Jahr) seine rewards abzuholen ehe sie verfallen
- Wenn eine "epoch" getriggert wird werden sofort alle rewards der letzten woche an die Voter ausgezahlt
- "recoverERC20" kann von "onlyOwner" genutzt werden um Tokens aus der Bribe zu nehmen die keine "rewardTokens" sind

## BribeFactory

- Die Bribe-Factory hat die Aufgabe, die Bribes zu erstellen
- Jeder Nutzer kann die BribeFactory über den Smart Contract ansteuern, da diese ohne die GaugeFactory nutzlos ist

## veTokens

- Der Zeitraum ist mitbestimmend über den Anteil, den man an Voting-Power erhält
- Beispiel: Werden STABLE / USDFI für 4 Jahre gelocked, erhält man den gleichen Anteil an veToken.
  - Bei 2 Jahren Locking erhält man nur noch die Hälfte der Voting-Power
  - Voting-Power erhält man durch das Locking von veTokens als Ausgleich für den vorher definierten Locking-Zeitraum ([veStable.vy](#))

## PairFactory

- Vergleichbar mit Gauges von Velodrom / Solidly.
- sAMM / vAMM
- AMMs können je nach LP einzeln die fees anpassen – von 0,01% bis zu 10%.
- Fees können in 3 Punkte eingeteilt werden: "80% USDFI-Maker", "20% LP Owner Bribes", "0% Partners" (weitere Punkte möglich, falls später nötig).
- LPs können mit „Max“ Gas-Preis beschränkt werden um frontrunning zu erschweren (der „Min“ Gas-Preis pro Chain darf nicht unterschritten werden)

## Besonderheiten

Wenn ein Gauge neu erstellt wird sollte zuerst darauf gevotet werden dann eingezahlt werden und dann ein kleiner betrag gebribed werden damit sie in der nächsten Woche einwandfrei läuft.