



Monitoring Broadcast

November 2023

TOP 5 Best Runners (October 8th - November 13th)

FXN: +199%	PENDLE: +76.6%	RPL: 73.6%	AAVE: +51.5%	ALCX: +27.9%
------------	----------------	------------	--------------	--------------

Convergence Eco

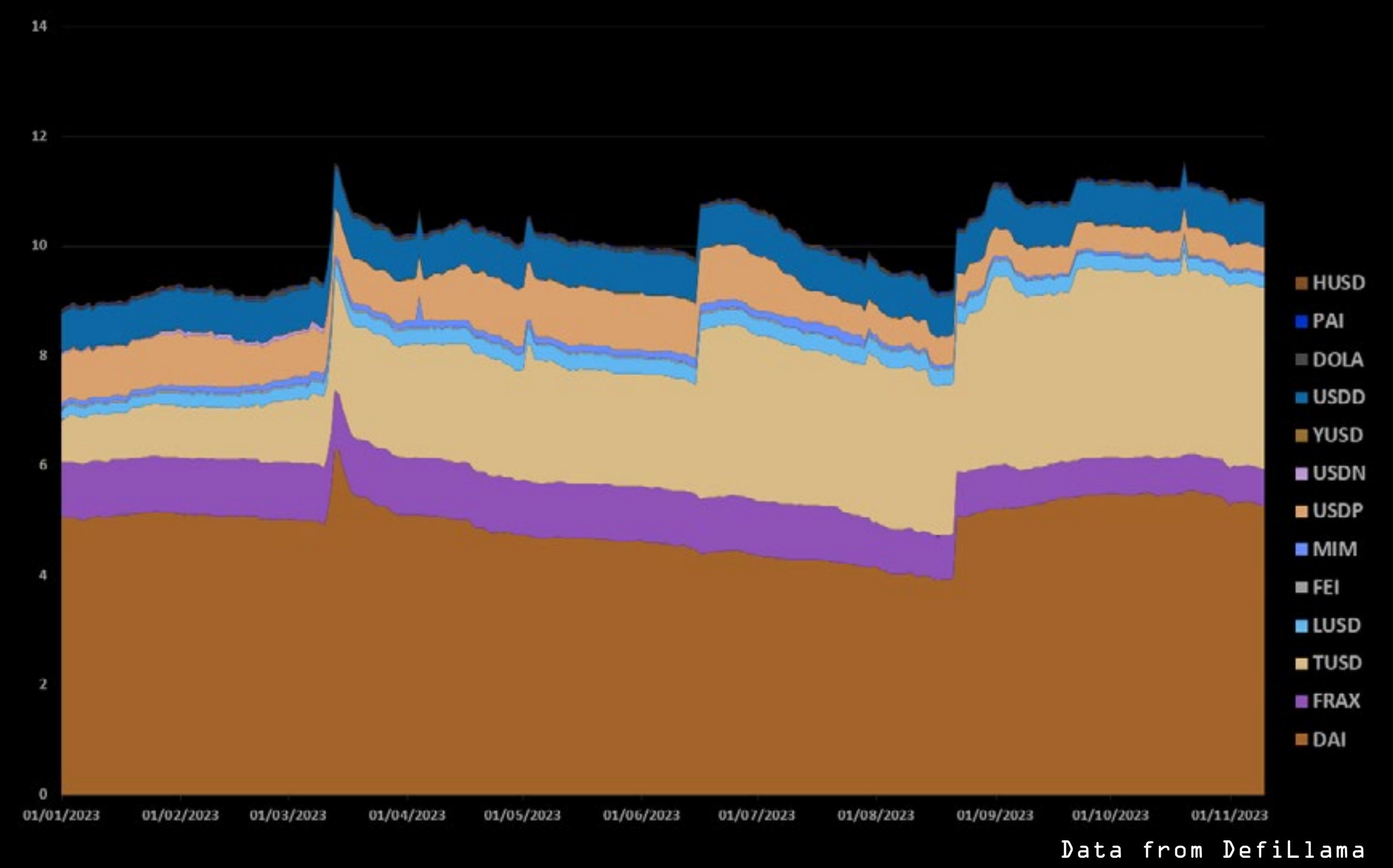
- The Convergence Proto DAO is officially live, marking a new dawn for decentralized, community-owned governance
- Three new gauges proposal were approved on Stake DAO
- cvgSDT will be the core of the StakeDAO integration, allowing users to boost their governance token yield through Convergence

- StakeDAO integration audit has started on Sherlock, an audit bounty platform, \$50,000 is dedicated to the contest
- Convergence held its first poker tournament, with an aim to become the Poker-house of the Curve ecosystem
- The dApp's landing page is currently being reworked to enhance the UX

Curve Eco

- crvUSD all time fees have exceeded \$1,800,000 showing quick and widespread adoption
- sFRAX consolidates its hype as deposits in the vault exceed \$40M. Potential trend for real-world assets (RWA)?
- Blackrock ETH ETF filling demand made the market soar, as volumes on-chain attain new local highs
- StakeDAO has launched 2 new Omnichain Fungible Tokens, asdCRV and sdMAV, using Layer0's OFT technology

DeFi-centric Stablecoin Dominance 2023 - (in billions)



Game Contest

This month, your observation skills will be tested. The rule is simple: find the five differences in each sets and win the prizes. Prizes are cummulative, don't blink!

1st solving: 1 cvgpepe
2nd and 3rd solving: 300 \$CVG



Grand Theft Crypto

```
function increaseLockTimeAndAmount(
  uint256 tokenId,
  uint256 durationAdd,
  uint256 amount,
  address operator
) external checkCompliance(tokenId, operator) onlyWalletOrWhiteListedContract {
  require(amount > 0, "LTC");
  ICvgControlTower _cvgControlTower = cvgControlTower;
  /// @dev retrieve actual staking cycle
  uint128 actualCycle = _cvgControlTower.cvgCycle();

  LockingPosition storage lockingPosition = lockingPositions[tokenId];
  uint256 oldEndCycle = lockingPosition.lastEndCycle + 1;
  uint256 newEndCycle = oldEndCycle + durationAdd;

  require(oldEndCycle < actualCycle, "LOCK_OVER");
  require(newEndCycle - actualCycle - 1 <= MAX_LOCK, "MAX_LOCK_96_CYCLES");
  require(durationAdd % TDE_DURATION == 0, "END_MUST_BE_TDE_MULTIPLE");

  if (lockingPosition.yPercentage != 0)
    uint256 _ysToReport = balanceOfYsCvgAt(tokenId, oldEndCycle - 1);
    totalSuppliesTracking[oldEndCycle].ysToAdd += _ysToReport;
    totalSuppliesTracking[newEndCycle].ysToSub += _ysToReport;

  _ysCvgCheckpoint(
```

```
function increaseLockTimeAndAmount(
  uint256 tokenId,
  uint256 durationAdd,
  uint256 amount,
  address operator
) external checkCompliance(tokenId, operator) onlyWalletOrWhiteListedContract {
  require(amount > 0, "LTC");
  ICvgControlTower _cvgControlTower = cvgControlTower;
  /// @dev retrieve actual staking cycle
  uint128 actualCycle = _cvgControlTower.cvgCycle();

  LockingPosition storage lockingPosition = lockingPositions[tokenId];
  uint256 oldEndCycle = lockingPosition.lastEndCycle + 1;
  uint256 newEndCycle = oldEndCycle + durationAdd;

  require(oldEndCycle > actualCycle, "LOCK_OVER");
  require(newEndCycle - actualCycle - 1 <= MAX_LOCK, "MAX_LOCK_96_CYCLES");
  require(durationAdd % TDE_DURATION == 0, "END_MUST_BE_TDE_MULTIPLE");

  if (lockingPosition.yPercentage != 0) {
    uint256 _ysToReport = balanceOfYsCvgAt(tokenId, oldEndCycle - 1);
    totalSuppliesTracking[oldEndCycle].ysToAdd += _ysToReport;
    totalSuppliesTracking[newEndCycle].ysToSub += _ysToReport;

    _ysCvgCheckpoint(
```

