Distribution: 100 USDC
Outstanding Supply: 1000 ACME

```
Jack  -  100  ~ 10%  ~ 10 USDC
Jerry -  250  ~ 25%  ~ 25 USDC
Sara  -  300  ~ 30%  ~ 30 USDC
Tom   -  150  ~ 15%  ~ 15 USDC
Fiona -  200  ~ 20%  ~ 20 USDC
```

```
Jack  - FAIL
Jerry - FAIL
Sara  - PASS
Tom   - PASS
Fiona - PASS
```

Deposit USDC Into Distribution Treasury

Option 1: Delegate Creates A Proposal to distribute

Option 2: Automatically distribute distributions

Run Identity Verification on cap table members

Identity Verification

At the time of distribution, we should calculate each user's percentage of the distribution. This poses a few tricky considerations such as partial distribution vs. ex-Dividend date that were caused by transferred LP tokens

Mint 35 Distribution Tokens

65 of distribution was streamed out and 35 remains, so 35 dollars will be left in the distribution treasury that can still be claimed

Fail

Pass — Transfer 30 USDC 30%

Pass — Transfer 15 USDC ~ 15%

Pass — Transfer 20 USDC ~ 20%

Transfer 25 Distribution Tokens ~ 25%

Transfer 10 Distribution Tokens ~ 10%

Jerry and Jack now have to come and go through KYC/AML in order to swap their distribution tokens for USDC

Swap Distribution Token for USDC

Run Identity Verification on cap table members

Identity Verified

Fail

Pass

Burn 25 Distribution Tokens / Transfer 25 USDC

Burn 10 Distribution Tokens / Transfer 10 USDC

Here we'll burn distribution tokens as the money comes out of the distribution treasury

Implementation without distribution tokens

Pros:

single point of logic for determining & issuing distribution amounts. Compartmentalizes distributions for any actor, protocol, or service

Design pattern is more compatible with protocols & services such as autocompounders, leverage, self-repaying loans, etc

Simplifies distribution – no need of a distribution token with redemption mechanics

Future-proof where overseers of the system are not responsible for transfers (on fees)

Self-service, reduced dependency on actors and overseers

Cons:

Claims logic is more complex, especially considering cases where users wait for multiple distribution events before claiming. Or cases where LP balance changes between claiming. It's implemented before with distributing rewards in eth DeFi

GP collects fiat

Distribution Treasury / Program

Provides Distribution Funds

Distribution Logic: calculate amount of USDC per LP token at the time of event

USDC Vault Holding Distributions

USDC gets claimed

Claims Logic: handles the transfers of USDC

Request USDC distributions

Other Protocols

User

KYC/AML

User

Opt-in by depositing LP tokens

Automated Streaming

USDC transferred

User

User

Protocols

User

Other TOKR Services (?)