TL;DR

Present to the community the next (and final) steps regarding the “rescue mission” of tokens sent by mistake to smart contracts of the Aave ecosystem.

Context

As described HERE, an item part of the Aave <> BGD engagement was to support users of the Aave ecosystem who sent by mistake tokens to the wrong smart contract addresses.

Holistically, this is a pretty extensive procedure, as it requires interacting with the majority of all smart contracts of Aave, via on-chain governance proposals.

Initially, we planned this “rescue mission” in 3 separate phases, each one of them described in the original post. Phase 1 (arguably the most complex and including the whole setup work) has been completed time ago, so only 2 and 3 were pending.

After evaluating all interactions required for these 2 phases, we decided to group them together in a single proposal.

Rescue Mission Phase 2&3

Scope of rescue

The tokens rescued in this final phase are the following:

- Tokens locked on Aave v1 Ethereum contracts.
- Tokens locked on Aave v2 (all networks) and Aave v2 AMM Ethereum contracts.
- Tokens locked on Aave v3.
- Only tokens with > $1’000 aggregated value are included, to not create more operational overhead on small amounts

.

We have made a snapshot of all the user addresses, assets, and amounts eligible for rescue in this rescue phase.

We recommend everybody with funds locked (on the current scope

) verify their address is present on the following documents, by just searching (Ctrl+f) their address:

https://github.com/bgd-labs/rescue-mission-phase-2-3/blob/main/js-scripts/maps/usersAmounts.json

or in the following dropdown:

List of claims per token

{

“0xd7D0882C9DFAa3610fee727FB2730c0302784127”: {

“V2_ETH_A_RAI”: “1481.16074087007480402 ethereum_v2_arai”

},

“0xaE697D26582aFb0c3503e214aA2F116D9f44a7c9”: {

“V1_ETH_A_BTC”: “1.92454215 ethereum_v1_awbtc”

},

“0x4556FE954B637BBb3F76b0C740248c23d832dA14”: {

“ETH_USDT”: “1010 ethereum_usdt”

},

“0x56DAa3C5f21C6b120d01428c513665a231CBfD57”: {

“ETH_USDT”: “600.057405 ethereum_usdt”

},

“0x5c9898D393f02d0C6c7332d42a94cfC4dF3D5100”: {

“ETH_USDT”: “20000 ethereum_usdt”

},

“0xd806826c605AAd41f243a94dDcA892Fd1526cfc7”: {

“ETH_USDT”: “10000 ethereum_usdt”

},

“0xEbA0C6DBE865F8e2ddC7FD26dAa395f822F23322”: {

“ETH_DAI”: “22000 ethereum_dai”

},

“0xaE261e8debc5f60F1560bf2CEC94e08aEE6934eb”: {

“ETH_GUSD”: “14881.06 ethereum_gusd”

},

“0xdACaa3F1f9F9CFC211F987D8F7E9E53691A4880a”: {

“ETH_GUSD”: “5113.8 ethereum_gusd”

},

“0x47Bf1B7934f4446B91a99D8BD79fC8c002E5360d”: {

“ETH_LINK”: “4084 ethereum_link”

},

“0x42CEb793F7Aa094b3Cd4B63C40D1E0af60c5BCBd”: {

“ETH_USDC”: “1 ethereum_usdc”

},

“0x4baD9Ef5a8011Dd5592b82B4DcB99E0C7cbcc145”: {

“ETH_USDC”: “1010.528332 ethereum_usdc”

},

“0xECeA185f14DeEaA708C9BE8Bf45B1798eb0B1938”: {

“ETH_USDC”: “78.361385 ethereum_usdc”

},

“0x58aAdb7fBf9Fb62e6A6462190B38B4eF95234F58”: {

“POL_WBTC”: “0.22994977 polygon_wbtc”

},

“0x517ed7f999806C767FcCF0f31dbF54dfe010977B”: {

“V2_POL_A_DAI”: “0.0001429561 polygon_v2_adai”

},

“0x6a93611Bcc572a67e60249C1D9F8F5F85ad50949”: {

“V2_POL_A_DAI”: “4220.580125141545600939 polygon_v2_adai”

},

“0xa1e29A54467debA55dbf27f8850bDE11466aB38E”: {

“V2_POL_A_DAI”: “30 polygon_v2_adai”

},

“0x1b63a8588eC051F84D99204F739ec4DCfd8E052E”: {

“V2_POL_A_USDC”: “97.869675 polygon_v2_ausdc”

},

“0x2222222229b89C7844F19ef503C4dC503Be47F84”: {

“V2_POL_A_USDC”: “0.000556 polygon_v2_ausdc”

},

“0x32622Cb41c18F2a9e63F205FF11d99620D654ebD”: {

“V2_POL_A_USDC”: “0.474647 polygon_v2_ausdc”

},

“0x72fbF878C35758e5629E0be3C26247402A3Cf681”: {

“V2_POL_A_USDC”: “0.000006 polygon_v2_ausdc”

},

“0xE01f36Eacd69414Cf187715Ebc46720f96E8103E”: {

“V2_POL_A_USDC”: “514033.033134 polygon_v2_ausdc”

},

“0x2E5EbbDeb2773b3632069068B044C5B116AC1936”: {

“POL_USDC”: “14.106892 polygon_usdc”

},

“0x4719b1FFF4499F41A14Bc098Cc440fba6E6a9Cd3”: {

“POL_USDC”: “2000 polygon_usdc”

},

“0x6139D4bbbf5c1280aF1EAA2BBD2Ffaae28E9033E”: {

“POL_USDC”: “0.136057 polygon_usdc”

},

“0x77B34173A36b01822B4Db9432C6511d11B01852D”: {

“POL_USDC”: “2500 polygon_usdc”

},

“0xe7d0977990B81cf52417D7964C8EB86e2793f8e7”: {

“POL_USDC”: “1 polygon_usdc”,

“AVA_USDC_E”: “1 avalanche_usdc.e”

},

“0x1f8b517e8E3CbB26125416a958Dbb44C7A5387d4”: {

“AVA_USDT_E”: “1772.206585 avalanche_usdt.e”

},

“0x358Dbc0B83186591F3202400ffCD70c14C1FF9aE”: {

“AVA_USDC_E”: “2521.408895 avalanche_usdc.e”

},

“0x3c82976E756E8280Ddb7AB9529E2063Ec1bBF90b”: {

“OPT_USDC”: “44428.421035 optimism_usdc”

}

}

Execution/next steps

Different from Phase 1, this Phase 2&3 will only require one governance proposal on Level 1 Executor (Short), given that all contracts affected are controlled by said Executor.

During the following days, once all the reviews are finished, we will submit a governance proposal with 3 payloads: 1 per network (Ethereum, Polygon, Optimism), containing the logic to release the locked tokens on Aave to the Rescue Mission smart contract

.

The payloads will also register the so-called “distributions” on the Rescue Mission contract, this way enabling anybody to do transactions to claim their funds from it.

Once/if the community approves the proposal, everybody eligible will be able to claim the locked tokens by interacting with the Rescue Mission smart contract, or via the utility interface available on https://rescue.bgdlabs.com/