# SOLIDIFIED

Audit Report for Polymath. March 14, 2018.

## Summary

This audit has been limited to the patcher applied to release 2.1 of the `CountTransferManager`, `GeneralTransferManager` and `ManualApproveTransfermanager` contracts of the Polymath protocol.

## Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the below token swap. The debrief took place on March 14, 2019 and the final results are presented here.

## Audited Files

- CountTransferManager.sol
- CountTransferManagerFactory.sol
- GeneralTransferManager.sol
- GeneralTransferManagerFactory.sol
- ITransferManager.sol
- ManualApprovalTransferManager.sol
- ManualApprovalTransferManagerFactory.sol
- GeneralTransferManagerStorage.sol

The files were audited on commit `a8b71e575526284c08803e156ab0c3feca198989` and solidity compiler version `0.4.24`

## Intended Behavior

The purpose of these contracts is to make general improvements to existing modules

## Issues Found

### Critical

No critical issues were found.

### Major

No major issues were found.

### Minor

#### 1. Conversion from uint256 to uint64 can overflow

In `GeneralTransferManager.sol` line 227 an `uint256` is converted to `uint64`, this leads to unexpected behavior when input value doesn't fit into the `uint64` range. This has a high likelihood to cause problems in the future. For example if some other contract validates input timestamp is larger than some value before passing them to `modifyWhitelist()` this validation can be bypassed by passing in a value that will overflow `uint64`.

**Recommendation:**
Either change the interface to match `uint64` storage format or store `uint256`.

**Amended [18.02.2019]**
The issue was fixed and is no longer present in commit `0008f283c1a6cd98555825faffb47979b79968d0`.

## Notes

### 2. Approvals array is unbounded

In `ManualApprovalTransferManager.sol` there's an unbounded array, which if grown too large it can make some operations impossible. We understand the need for that array to be able to have the `getActiveApprovalsToUser()` function, but it's tradeoffs must be further analyzed.

**Polymath's response:**
This array is only used within a getter (`view`) function which will be called by an off-chain process, so gas costs are not relevant.

**Amended[18.02.2019]**
Solidified agrees with Polymath's and consider this issue fixed.

### 3. Sanitize inputs

In `GenerealTransferManager.sol`, the functions `changeDefault`, `changeIssuanceAddress` and `changeSigningAddress` don't perform any kind of input sanitization. It could be useful to prevent some invalid inputs.

**Amended [18.02.2019]**
The issue was fixed and is no longer present in commit `0008f283c1a6cd98555825faffb47979b79968d0`.

### 4. ModifyManualApproval has confusing type

`ManualApprovalTransferManager.sol`: the use of integer for `modifyManualApproval` for change direction is confusing. Consider replacing it for a `bool` or an `enum`

**Amended [18.02.2019]**
The issue was fixed and is no longer present in commit `0008f283c1a6cd98555825faffb47979b79968d0`.

## 5. Smaller gas optimizations

ManualApprovalTransferManager:
* Line 199, 200: could be merged into one variable to safe gas
* Line 216: could be removed
* Redundant storage reads in lines 86,87,125,126,266,269,358 and 359. Those could be optmized to perform only one storage read:

```
uint256 index = approvalIndex[_from][_to] - 1;
      if (index != uint256(-1)) {
...
}
```

**Amended [18.02.2019]**
The issue was fixed and is no longer present in commit `0008f283c1a6cd98555825faffb47979b79968d0`.

## 6. Improvements on clarity

Consider renaming variables `toTime` and `fromTime` to `receiveTime` and `sendTime`. This greatly improves clarity, and will diminish the time one needs to fully understand the smart contract.

**Amended [18.02.2019]**
The issue was fixed and is no longer present in commit `0008f283c1a6cd98555825faffb47979b79968d0`.

## Closing Summary

Although no critical or major issues were found, there are some minor issues that can affect the desired behavior and it's recommended that they're addressed before proceeding to deployment.

## Disclaimer