Morpher Audit Result

Audit Scope + Non-scope

- On March 2nd, 2020 Martin Froehler walked us through the Morpher system and we extracted a list of claims that the smart contracts should uphold.
- From March 2nd until March 6th the Capacity Team worked on assessing the code and compiling a first review based on an early preview.
- From April 20th until April 24th the Capacity Team worked on assessing the code and compiling this document.
- We checked only solidity code and its associated tests, no backend systems or web3 code.
- Interaction between different chains was out of scope for this audit, we only checked interactions on the sidechain.

Audit performed on the following material:

Repository: Morpher-io/MorpherProtocol
Commit: `3090075bdd3a6c866d067c7c8c2164262497374f`
Commit Date: Mon. Apr 20 14:24:19 2020 +0200

Filename	Hash (SHA256)
IERC20.sol	f0d2e91d6777acce30eeb5a925daa5a340f2ada67cc699eee68390e8b9646c6e
IMorpherToken.sol	20e6e2dab141d2b7f4932a40fa2611b33801999b7caf0a98fb85d65b5bd05a46
MerkleProof.sol	1648f54d505b82a03943013b51ea9fddcd1e5bf8024f4c889fbf8ab40b71ecec
Migrations.sol	d5b76f6c454eb4e883e161e92ad6db9dd6b0b8c7893bb6ee6044de22bb0f633e
MorpherAirdrop.sol	22b412c2e8a0d21125c09c17cf28c52722812406cf56e474a65421612d2f2484
MorpherBridge.sol	da3d1af6f601cfd717a37c865e489b41525a9fd57f36683c39d0c594ecf09438
MorpherEscrow.sol	6bbd55f6c131b18843e6fd6db1ab5feeb6543bae6bb6b760bcfeacbbc8505348
MorpherGovernance.sol	d04f214a516e45ee65f9d7e34c6a43dc8710c869ff6a5de372d0ba63f798ae80
MorpherOracle.sol	7caaab41e1adc8de0e80fef8cdf0c4e74819f3d999d238f39c8a7bf14aea2a6e
MorpherState.sol	1f171b9a114e63a990c450954685fef3c6522f2471fdbacc0cdb8c82fe91b580
MorpherToken.sol	658f8e023af34f5b713208ef9528410eca3da5cfb5a2e1eb3db3aa6564d53112
MorpherTradeEngine.sol	0350f5a41f952c34d852ae824fc945bf419db20d494b3b676bbd5570fdbd30a1
Ownable.sol	40fc1a86d4797e93ee7398f31027e3e4e85862f09445fb2b9254baca3e374f03
SafeMath.sol	41a9565e5e2d263fe470c777c19270311714b8107c8676c822a13fcddb070a1e

Overall Summary and recommendations

Other than an easily fixable compile issue, we have not identified any significant bugs or vulnerabilities in the contracts themselves and would not see an issue if the system was already deployed in its current state.

That said, we recommend some additional improvements to ensure state access is only granted to the correct addresses (not not for example the deployer as in the current migration scripts) as well as some efficiency and readability improvements.

Additionally, we recommend that after deployment (but before the whole system going live), the assignment of the state contract ownership to a cold storage wallet and sensible assignment of state access is being verified by an external party.

Suggested development process improvements

- `truffle-config.js` should use a specific compiler version.
- Improve automated tests with significant coverage (90%+) especially for error conditions, not just the mainline functionality.

Identified Issues independent of specific claims

Functional Issues

- While we were told that replacing the state contract is intended to be possible as something like a “hard fork” scenario, the functionality for this is not implemented consistently. A possible solution to synchronize a potential state contract switch would be to have a single reference holder contract that is deployed as the first contract and whose address is given as a dependency to all state-accessing contracts.
- `SafeMath.sol` doesn’t compile - it includes functionality from OpenZeppelin 3.0 but wrongly ported (require() with 2 error strings does not exist). This was trivially fixed to enable the audit.

Functional Recommendations

Morpherstate.sol

- If the trade engine address would be stored in the state, the whole `stateAccess` mechanism (and the `onlyPlatform` modifier) could be replace with naming the address that have access explicitly via the already stored variables for them, which would make it more transparent who it actually is that gets access to those state functions.

Runtime Efficiency issues

- Sending `now` or `block.timestamp` as explicit argument in events is not useful as any event consumer gets the linked block and transaction info along with the event itself automatically.

MorpherAirdrop.sol

- `setAirdropAuthorized()` has a return value but doesn't need one as it's always `true`.
- The `require()` in `claimAirdrop()` and `adminSendAirdrop()` is not actually required as both the `.sub()` would revert as well and `_sendAirdrop()` also does the same check.

MorpherGovernance.sol

- `countVotes` should not be a contract-wide state variable but instead local to the functions it's used in (probably as a fixed-size memory array) - which removes the need for zeroing out its values and saves gas for s state variable.

MorpherOracle.sol

- The timestamps on the events are not strictly necessary, as anyone listening to events can always read them from the block data associated with the event.

Morpher State.sol

- `getAllowance()` exactly replicates the already existing automatic getter on the `mapping public allowed`, it would be better to reduce bytecode output and either eliminate the extra getter or set the `mapping` to `private` instead. A similar question comes up with `balances` and `balanceOf()`, with the getters and `address` variables for various platform roles, and some other getters as well.

MorpherToken.sol

- The `"transfer amount exceeds balance" require()` in `_transfer()` is technically not needed as the `.sub()` in the state will revert anyhow. That said, it may add more transparency for code readers to have it in the token contract.
- The contract is `Ownable` but the owner has no actual function in the contract.

Cosmetic/Code stylistic issues and pattern consistencies

- Contracts from external sources like in this case OpenZeppelin (`MerkleProof`, `Ownable`, `SafeMath`) should probably be put into their own subdirectory to signify they are not self-developed code. If that external source is not installed by dependencies, version of the source should be provided alongside the code taken from there.
- A pragma like "`^0.5.11`" would be preferable to the "`>=0.4.25 <0.6.0`" used now as there is no reason to allow 0.4.x at all and the "`^`" syntax means "same in first two parts, greater or equal for the third part of the version".
- Deployment script in migrations currently only works for network "local", it should be made to work with the live network(s) that are the target for deployment, so that the migration scripts serve as a safety net to do the deployment correctly and as documentation of what was actually done for deployment.

- Test process improvement: It would be a good idea to start ganache with a mnemonic for testing (e.g. have a start_ganache script for that) and use the different addresses of the mnemonic for the different roles in tests as well so you can verify that transfers of roles in contracts actually work.
- Test process improvement: You should not define your own “web3” variable in migrations at all, as truffle already injects its own “web3” variable in the global namespace, which you can just use. That also means you do not have to define host or port of the Ethereum node a second time in there.

MorpherAirdrop.sol

- The mappings `airdropClaimed` and `airdropAuthorized` should have visibility set explicitly - if it was set to `public`, the `getAirdropClaimed()` and `getAirdropAuthorized()` getter functions could be removed and the automatic ones be used instead - but even if the mappings stay `private`, that should be noted explicitly.

MorpherBridge.sol

- The constant `ONEDAY` is only used once and just set to “`1 day`”. Everything surrounding that place also talks about “24 hours” specifically, I it may make sense to just directly use a literal “`1 day`” there in the function and not define the constant at all.
- The contract has a number of functions that have no `modifer`/`require` restricting then to side chain or main chain, but with a name implying a certain chain, which can be confusing, e.g. what’s happening when I call `transferToSideChain()` on the side chain’s bridge contract? Is that intended to work? Which functionalities are actually intended to be the same on both directions?
- The function disableFastTranfers has a typo (disableFastTransfers)

MorpherEscrow.sol

- The comments about “`Change to 30 days after testing`” could be removed as that is set as a constant now anyhow.

MorpherGovernance.sol

- `MAXVALIDATORS` is set somewhat high (to 99) as the for loop may run out of gas with way less than that, and the token supply may not be enough to even get to that limit either - the initial total supply cannot support more than 13 validators. It may be a good idea to run some tests on gas costs implied for voting when adding another validator, and calculate a more realistic max number from that and/or (estimated) available token supply.
- The contract has a few getters for public mappings that are redundant with the automatically generated ones. See comments on `MorpherState.sol` about this topic.

MorpherOracle.sol

- In `cancelOrder()`, both sides of the `if` condition end up doing the same and the `if` can be removed.
- In `createOrder()` it may make sense to handle potentially sent currency (send it back or forward it somewhere) when the gas price is zero.

MorpherState.sol

- `IERC20.sol` is imported but never actually used in the contract.

MorpherTradeEngine.sol

- The `mapping orders` should specify visibility explicitly, even if that is `private`.
- The lines in `processBuyOrder()` and `processSellOrder()` are way too long to read and call functions multiple times. Line breaks should be added for readability,
- Results of calls like `shortShareValue()` and `state.getShortShares()` (and well as their long variants) should be stored in function-local variables and re-used in the various places they are needed.
- With those readability changes, the `buyIt()` and `sellIt()` functions could potentially be inlined into the above functions as they are only used there.
- In `longShareValue()`'s line 476, it would be better understandable and match the comment if you first do the `.div(PRECISION)` and then just `.sub(1)` to match the "`minus one`" in the comment.
- In line 478 (same function), the additional brackets around `.mul().div()` are unnecessary. That would even be true for normal arithmetic operators, but those `SafeMath` operators work in a chain anyhow, so e.g. *(1+2)*3* would just be `1.add(2).mul(3)` which returns *9*, and *1+2*3* would be `1.add(2.mul(3))` and return *7*.

Claims

Claims which were tested that hold (OK)

— **"The Morpher Protocol does not repeat the same known mistakes in contract design as previously discovered in dForce, TheDAO, Parity Multisig, KickICO"**

- As noted in https://decrypt.co/26033/dforce-lendfme-defi-hack-25m the dForce hack is supposedly the same as an already-known vulnerability of Uniswap with ERC777, which uses reentrancy. Due to Morpher not calling external contracts (which could be controlled by an attacker), reentrancy in this form is not possible, so the attack should not be possible.
- TheDAO was attacked via a reentrancy, closely described in https://medium.com/@zhongqiangc/smart-contract-reentrancy-thedao-f2da1d25180c - The Morpher contracts do not call external contracts (which could be attacker-controlled), so reentrancy is not an issue.
- The attack on Parity Multisig wallets was achieved by delegating calls to a library without any safety checks that the respective calls are safe to execute in this setting, the details are explained for example in https://blog.openzeppelin.com/on-the-parity-wallet-multisig-hack-405a8c12e8f7/ - Morpher does not use libraries or `delegatecall` and so should be safe from this pattern.
- The KickICO compromise was a hacker gaining access to the private key of the owner of the token contract (also the contract deployer) as described in https://medium.com/@zhongqiangc/analysis-on-kickico-hack-part-1-a09e99de2480 - in addition, that owner could burn and mint arbitrary amounts of tokens owned by any account. In the Morpher contracts, only addresses with

platform access can arbitrarily burn and mint like that, but the owner has control over which addresses get this access. As stated in claims, the state owner needs to be a cold storage address.

— **"The token follows ERC20 specs and expectations"**

- The initial minting in the `MopherState constructor` does not emit `Transfer()` events on the token contract. This should probably be done with the same code as in the `mint()` function. A good pattern to use is to have an internal `_mint()` function that both places call.
- By looking through `Transfer` events, the `totalOnOtherChain` and `totalInPositions` amounts are never seen and so `totalSupply()` does not match the balances seen from the outside. It would be better if there would be placeholder account addresses representing/holding those amounts, e.g. `0x1` and `0x2`, preferably set as constants in the state contract. That way, updating those balances could be done by using the standard `mint/burn` mechanics and the `totalSupply` could be adapted fully automatically, and code could be simplified by not having separate variables and not needing the `totalToken` variable and usage of `updateTotalSupply()`.
- Any user can burn tokens they own. This is completely fine, just a property of the tokens that should be noted.
- Minting tokens to the `0x0` address is technically allowed, would be good to be blocked with a `require()`.
- It would be good to include an independent suite like https://github.com/CryptoverseRocks/token-test-suite in the tests to make sure ERC20 support works as expected (unfortunately, this suite is unmaintained and has some bugs, but those can be straightened out).
- After some fiddling with test setup and fixing the test suite itself, token-test-suite PASSES!

— **"Users will receive Tokens via Airdrop"**

- There is no automatic airdrop functionality, but the MorpherAirdrop contract has all necessary functionality for the airdrop admin to spread it manually (and recipients to claim it).
- By code inspection, the claim holds, as long as the airdrop contract actually gets tokens assigned manually.

— **Oracle is a trusted entity and leads to a proportionate increase/decrease of Morpher tokens"**

- Oracle contract can be set (by governance) in state but has no specific privileges there and therefore is not trusted to take actions there directly.
- Oracle contract can call `requestOrderId()`, `processOrder()`, `userCancelOrder()` on the trade engine.
- Of those, only `processOrder()` ends up modifying token amounts, as it ends up in `setPositionInState()` which calls `mint()` or `burn()` on the

state with amounts calculated in proportion to the order value data as given by the Oracle.

- Calling `createOrder()` on the oracle contract opens an order in the name of the caller and in the case of mainnet requires a payment of gas costs for the callback along with that call. Any amount above a certain limit will be ingested and those funds are completely forwarded to the `callBackCollectionAddress`, regardless of over-payment. In case of the side chain, where no gas is needed, the oracle contract keeps any potential currency sent along with this call.
- The actual processing operation is initiated by an external address calling the callback. The owner of the Oracle contract can enable multiple such addresses at will and they have no outside-visible relation to the above-mentioned callback address.
- With that, the claim holds with multiple limitations: the oracle contract is trusted only within the trade engine, the values it returns directly affect how positions will be added, removed or liquidated and consequently tokens for a user will be added or removed (outside of the liquidation case, which results in a loss of all assets based on a single value delivered by the oracle). All those consequences only affect positions/tokens matching orders that have been created by users via the oracle contract and not processed before. The owner of the oracle contract has full power over who can inject those values and call the callback, and who gets funds sent to supposedly cover gas costs for callbacks.

— **“Latency of market data < block time prevents frontrunning”**

- Assuming the Oracle operator(s) being trustworthy and sending correct data, this is pretty much guaranteed by the two-step process of first the users requesting an order, then the Oracle operator(s) looking up current data and calling back with that to process the order. If the market data in the callback is from a point *at or after* the user requesting the order, no front running is possible unless the user has information from the future.
- With that, the claim holds.

— **“if connected to the sidechain and equipped with sidechain ether users can directly interact with the trading system - without any access to internal morpher databases”**

- Users can create orders and cancel their orders on the trading system via the oracle contract. They can also view their own orders and any positions directly via the trade engine contract.
- The claim holds when taking the Oracle contract as part of “the trading system”.

— **“A trading history can be reconstructed by looking at the event log.”**

- The oracle contract defines and emits events for `OrderCreated`, `LiquidationOrderCreated`, `OrderProcessed`, `OrderFailed`, and `OrderCancelled`.

- The trade engine contract defines and emits `PositionLiquidated`, `OrderCancelled`, `OrderIdRequested`, `OrderProcessed` and `PositionUpdated` events.
- The contents of the events look to be complete.
- The claim holds for the event logs of both the oracle contract and the trade engine.

— **"Governance can vote for oracle contract and set administrator"**
- See claim of *"if users are not satisfied …"*, that claim holds.

— **"Administrator can set total supply of cash + positions"**
- Using `setTotalInPositions()` and `setTotalOnOtherChain()`, the administrators can set arbitrary values for those two pieces. Both functions call `updateTotalSupply`, which adapts `totalSupply` to reflect those changes.
- The claim holds though it could even be improved upon, see notes in the ERC20 claim.

— **"Admin can activate/deactivate markets"**
- `activateMarket()` and `deActivateMarket()` allow doing this and are gated to only the administrator, so the claim holds.

— **"Admin can set leverage"**
- That claim holds as well via the `onlyAdministrator`-gated `setMaximumLeverage()` function.

— **"Admin can pause tokens transfers and restrict the right of the users to transfers"**
- `pauseState()` and `unPauseState()` are `onlyAdministrator`-gated functions and the variable they control blocks any token transfers including minting and burning - which also includes any trade that changes token balances.
- The claim holds with the side-effect of also stopping all trade.

— **"Damage is constrained on a single market in case of liquidation - your remaining balance and positions are not affected."**
- As liquidation is only called when processing a single order, and only acts on that one position for the affected address and market, the claim holds.

— **"If users are not satisfied, they can replace the oracle and protocol administrator with another provider - this is done through a voting through simple majority procedure."**
- Anyone can become a validator, which removes an increasing amount of tokens from their wallet based on how many previous validators exist.
- After a warm-up period (constant set to 7 days) after they themselves or the most recent validator joined, any validator can place a vote for oracle or

administrator, which triggers an election right away. No checks are done if the given address supports oracle functionality.
    - Anyone can trigger an election, any votes given until then will be counted and a simple majority for one oracle or administrator wins. On vote equality, the winner is whoever that got to that number first when evaluating the validators in order. The election directly sets the given address as oracle or administrator.
    - Validators can step down and 99% of the tokens used for becoming a validator are returned (and partial 99% sums for validators that joined later).
    - Large enough token holders can block an election for another warm-up period ny joining as a validator, but that doesn't stop the possibility, just delay it. They can also not lengthen the block by repeatedly stepping up and down as validators, as stepping down immediately triggers a vote among the remaining validators.
    - The claim holds for the oracle contract and the administrator (which can (un)pause the token and trade system, (de)activate markets, and set maximum leverage as well as total position values).

— **"The token balance of Non-trading users can not be manipulated"**
    - If the state owner can be manipulated to grant state access to an address of a bad player, those could transfer, mint and burn tokens arbitrarily.
    - Otherwise, only the user can use transfer on the token to send tokens somewhere or burn their own tokens.
    - Bridge: any user can withdraw tokens from the chain (which get burned in exchange for a hash to be used as part of a merkle proof when restoring on the other side), any user can claim tokens (which get minted) with a fitting merkle proof, the bridge operator can enact such a claim with fewer limitations.
    - Governance transfers tokens to itself when becoming a validator and returns 99% of those when stepping down, burning the rest.
    - The claim holds wrt balances only being changed, in intended ways, by direct user interaction, and by trade.

Claims which were tested that don't hold under all circumstances (Maybe OK)

— **"Real time market data injected via oracle"**
    - Via the callback, the Oracle operator(s) inject data about current price, spread and potential liquidation into the system. People need to trust that this represents current market data.
    - The claim is unclear as there is no assurance by Smart Contract that data is believable and the assumption only holds if the operator(s) can be trusted to not send wrong data or have bad intentions.

— **“Token balances and positions held on a sidechain can be reclaimed / reclaimed on the main chain if the sidechain ceases operation for more than 72 hours.”**
    - If the owner (not the side chain operator!) of the “mainnet” bridge contract does not call `updateSideChainMerkleRoot()` for more than the `inactivityPeriod` (default 3 days), `recoverPositionFromSideChain()` and `recoverTokenFromSideChain()` become callable by anyone. Together with a merkle proof leading up to the last available root as given with the former function, and a leaf matching the sender and given position or balance, that info will be added to the contracts on the “main” chain.
    - In the case of token transfers, an additional limit of overall withdrawals per day is enforced.
    - The claim holds only if there is a way to access the merkle proof in a trustless way and with the limitation of the daily token withdrawal limit.
— **“The maximum damage, even in case of backend system breach is limited to the position amount”**
    - Any address/contract with state access can transfer, burn and mint tokens at will, and the owner of the state contract can assign state access to anyone. The migration script for deployment specifies for the token, trade engine, bridge, governance and contract deployer (!) to get such access (there is no automation for that). The idea is that the owner of the state contract will be set to a cold storage wallet, which mitigates that mostly if only trusted addresses have been granted access before that and nobody can be tricked to use the cold storage address to grant access to someone unwanted. It would be possible to close this issue almost entirely if state access would be gated to the already-stored-in-state contract addresses. Also see comments on `MorpherState.sol` about this topic.
    - The side chain operator can create tokens via the bridge on both sides by sending bogus values to the bridge contracts.
    - The oracle operator can only affect positions of orders a user has actually created.
    - So, if “backend system” only means “oracle operator”, the claim holds, but if it includes the side chain operator or even unspecified addresses with state access, the claim fails.

Claims which were tested that don’t currently hold (Not OK)

— **“State owner is a cold storage wallet and can add more permissions. only after the beta this account will be set to 0x0 address”**
    - Probably for practical reasons, the current migration scripts do not contain transfer of this address to a cold storage wallet.
    - It’s impossible to verify the part of the claim that this will be set to 0x0 as for verification this would need to be in migrations or have already happened. We can verify that it is possible to do so, but it would take away the possibility to

replace contracts in the system as well as to change reward address or margin.

- The claim fails the way it's currently phrased.
  It may be advisable to only keep the "cold storage" portion of this claim and actually have that audited/verified after actual deployment. On renouncing ownership, it can be audited that it's possible but verifying it actually being done would be making a prediction. An intent of that can be stated, but as long as upgradability of contracts and controlling rewards is valuable, renouncing ownership would stand in contradiction to that.

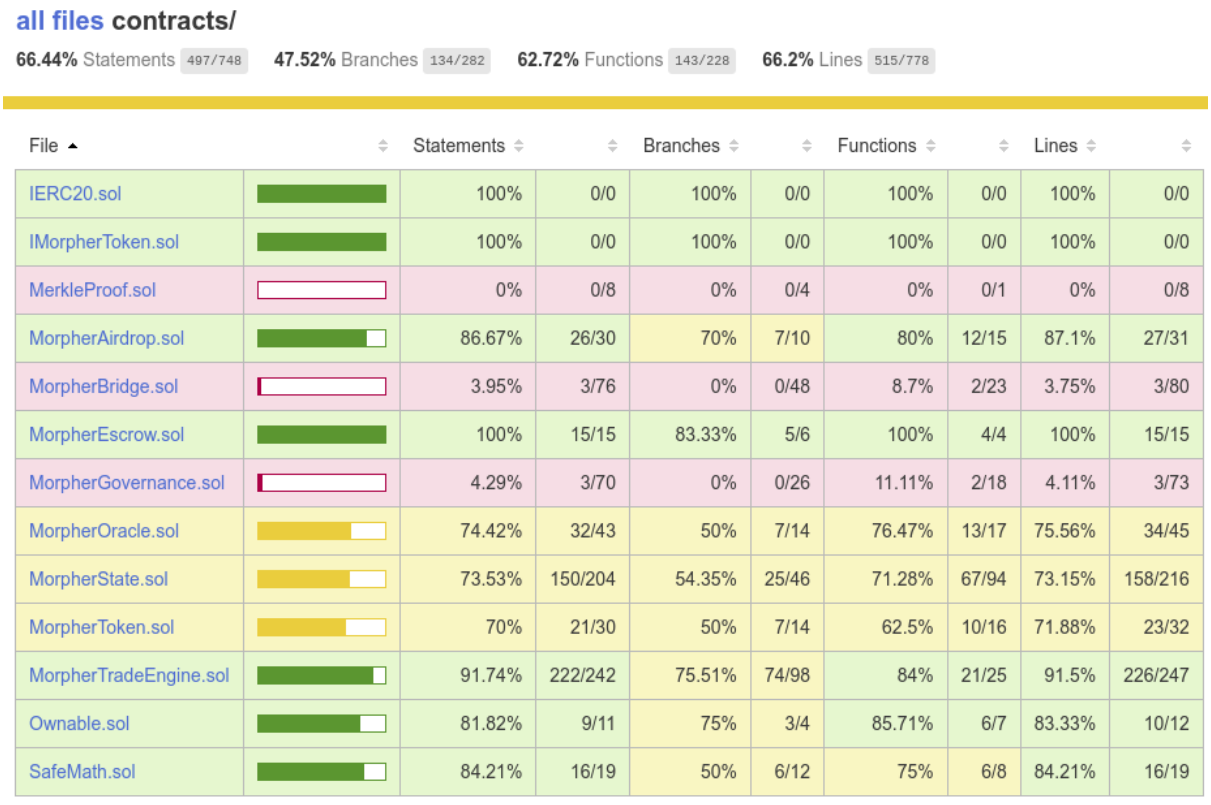
Automated Tools

Our Summary from automated tools

We attempted to run the following automated analysis tools on the contracts; Solidity Coverage, EthLint.

Below are some of these results.

- **Solidity Coverage**
  - Overall test coverage is not ideal (also factors in OpenZeppelin libraries as they have just been dropped into the same directory)
    - **66.44%** Statements `497/748`
    - **47.52%** Branches `134/282`
    - **62.72%** Functions `143/228`
    - **66.2%** Lines `515/778`



all files contracts/

66.44% Statements 497/748 **47.52%** Branches 134/282 **62.72%** Functions 143/228 **66.2%** Lines 515/778

File	Statements		Branches		Functions		Lines	
IERC20.sol	100%	0/0	100%	0/0	100%	0/0	100%	0/0
IMorpherToken.sol	100%	0/0	100%	0/0	100%	0/0	100%	0/0
MerkleProof.sol	0%	0/8	0%	0/4	0%	0/1	0%	0/8
MorpherAirdrop.sol	86.67%	26/30	70%	7/10	80%	12/15	87.1%	27/31
MorpherBridge.sol	3.95%	3/76	0%	0/48	8.7%	2/23	3.75%	3/80
MorpherEscrow.sol	100%	15/15	83.33%	5/6	100%	4/4	100%	15/15
MorpherGovernance.sol	4.29%	3/70	0%	0/26	11.11%	2/18	4.11%	3/73
MorpherOracle.sol	74.42%	32/43	50%	7/14	76.47%	13/17	75.56%	34/45
MorpherState.sol	73.53%	150/204	54.35%	25/46	71.28%	67/94	73.15%	158/216
MorpherToken.sol	70%	21/30	50%	7/14	62.5%	10/16	71.88%	23/32
MorpherTradeEngine.sol	91.74%	222/242	75.51%	74/98	84%	21/25	91.5%	226/247
Ownable.sol	81.82%	9/11	75%	3/4	85.71%	6/7	83.33%	10/12
SafeMath.sol	84.21%	16/19	50%	6/12	75%	6/8	84.21%	16/19

Code coverage generated by istanbul at Mon Apr 20 2020 13:54:35 GMT+0200 (Central European Summer Time)

- **EthLint**
  - “Avoid using now/block.timestamp” can be ignored
  - Otherwise mostly comments on indentation, line length, trailing whitespace, missing require error messages

```
> solium -d contracts/

contracts/MorpherAirdrop.sol
  9:1       warning     Line contains trailing whitespace              no-trailing-whitespace
  21:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  24:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  31:4      warning     Line exceeds the limit of 145 characters       max-len
  31:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  32:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  33:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  37:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  80:91     warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  92:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  99:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  120:8     warning     Line exceeds the limit of 145 characters       max-len
  124:108   warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members

contracts/MorpherBridge.sol
  50:8      warning     Line exceeds the limit of 145 characters                                                max-len
  50:16     warning     Avoid using 'now' (alias to 'block.timestamp').                                         security/no-block-members
  53:4      warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  60:102    warning     'require': The last argument must not be succeeded by any whitespace or comments (only ')'). function-whitespace
  63:4      warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  110:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  111:12    warning     Avoid using 'now' (alias to 'block.timestamp').                                         security/no-block-members
  112:34    warning     Avoid using 'now' (alias to 'block.timestamp').                                         security/no-block-members
  115:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  143:16    warning     Avoid using 'now' (alias to 'block.timestamp').                                         security/no-block-members
  156:4     warning     Line exceeds the limit of 145 characters                                                max-len
  158:8     warning     Line exceeds the limit of 145 characters                                                max-len
  159:8     warning     Line exceeds the limit of 145 characters                                                max-len
  163:4     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  172:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  174:8     warning     Line exceeds the limit of 145 characters                                                max-len
  175:5     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  176:3     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  179:4     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  182:1     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  183:1     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  190:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  202:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  203:8     warning     Line exceeds the limit of 145 characters                                                max-len
  203:16    warning     Avoid using 'now' (alias to 'block.timestamp').                                         security/no-block-members
  204:8     warning     Line exceeds the limit of 145 characters                                                max-len
  205:8     warning     Line exceeds the limit of 145 characters                                                max-len
  207:8     warning     Line contains trailing whitespace                                                       no-trailing-whitespace
  233:8     warning     Line exceeds the limit of 145 characters                                                max-len
  237:8     warning     Line exceeds the limit of 145 characters                                                max-len

contracts/MorpherEscrow.sol
  27:33     warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  50:12     warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  59:96     warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members

contracts/MorpherGovernance.sol
  29:4      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  54:8      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  56:4      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  112:44    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  113:30    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  124:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  125:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  129:8     warning     Line exceeds the limit of 145 characters                             max-len
  153:8     warning     Line exceeds the limit of 145 characters                             max-len
  153:79    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  154:8     warning     Line exceeds the limit of 145 characters                             max-len
  154:65    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  162:8     warning     Line exceeds the limit of 145 characters                             max-len
  162:79    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  163:8     warning     Line exceeds the limit of 145 characters                             max-len
  163:65    warning     Avoid using 'now' (alias to 'block.timestamp').                      security/no-block-members
  191:13    warning     Assignment operator must have exactly single space on both sides of it. operator-whitespace

contracts/MorpherOracle.sol
  20:3      warning     Line contains trailing whitespace              no-trailing-whitespace
  23:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  25:4      warning     Line contains trailing whitespace              no-trailing-whitespace
  38:0      warning     Only use indent of 4 spaces.                   indentation
  39:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  46:0      warning     Only use indent of 4 spaces.                   indentation
  47:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  61:0      warning     Only use indent of 4 spaces.                   indentation
  62:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  72:0      warning     Only use indent of 4 spaces.                   indentation
  73:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  78:0      warning     Only use indent of 4 spaces.                   indentation
  83:0      warning     Only use indent of 4 spaces.                   indentation
  88:0      warning     Only use indent of 4 spaces.                   indentation
  91:6      warning     Line contains trailing whitespace              no-trailing-whitespace
  93:0      warning     Only use indent of 4 spaces.                   indentation
  94:8      warning     Line contains trailing whitespace              no-trailing-whitespace
  98:0      warning     Only use indent of 4 spaces.                   indentation
  103:0     warning     Only use indent of 4 spaces.                   indentation
  108:0     warning     Only use indent of 4 spaces.                   indentation
  119:4     warning     Line contains trailing whitespace              no-trailing-whitespace
  120:3     warning     Line exceeds the limit of 145 characters       max-len
  120:3     warning     Only use indent of 4 spaces.                   indentation
  125:1     warning     Line contains trailing whitespace              no-trailing-whitespace
  134:39    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  139:48    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  144:46    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  149:4     warning     Line contains trailing whitespace              no-trailing-whitespace
  149:47    warning     Avoid using 'block.timestamp'.                 security/no-block-members
```

```
  158:55    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  199:10    warning     Line contains trailing whitespace              no-trailing-whitespace
  210:12    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  228:12    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  237:32    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  242:4     warning     Line contains trailing whitespace              no-trailing-whitespace
  242:33    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  259:80    warning     Avoid using 'block.timestamp'.                 security/no-block-members
  274:205   warning     Line exceeds the limit of 145 characters       max-len
  295:12    warning     Avoid using 'block.timestamp'.                 security/no-block-members

contracts/MorpherState.sol
  211:26    warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  285:47    warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  308:41    warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  603:43    warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members
  802:12    warning     Avoid using 'now' (alias to 'block.timestamp'). security/no-block-members

contracts/MorpherToken.sol
  40:4      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  80:1      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  169:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  171:5     warning     Only use indent of 4 spaces.                                         indentation
  184:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  186:5     warning     Only use indent of 4 spaces.                                         indentation
  204:89    warning     Visibility modifier "internal" should come before other modifiers.   visibility-first

contracts/MorpherTradeEngine.sol
  86:4      warning     Line contains trailing whitespace                                    no-trailing-whitespace
  114:4     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  119:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  163:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  163:16    warning     Only use indent of 12 spaces.                                        indentation
  164:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  197:12    warning     Avoid using 'block.timestamp'.                                       security/no-block-members
  225:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  226:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  227:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  228:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  229:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  232:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  233:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  234:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  235:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  236:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  288:20    warning     Avoid using 'block.timestamp'.                                       security/no-block-members
  310:20    warning     Avoid using 'block.timestamp'.                                       security/no-block-members
  352:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  353:2     warning     Only use indent of 8 spaces.                                         indentation
  359:1     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  390:8     warning     Line exceeds the limit of 145 characters                             max-len
  393:48    warning     Avoid using 'block.timestamp'.                                       security/no-block-members
  423:9     warning     Only use indent of 12 spaces.                                        indentation
  468:9     warning     Only use indent of 12 spaces.                                        indentation
  508:252   warning     There should be no whitespace or comments before the semicolon.      semicolon-whitespace
  591:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  595:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  607:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  609:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  614:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  619:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  620:8     warning     Line exceeds the limit of 145 characters                             max-len
  635:5     warning     Only use indent of 4 spaces.                                         indentation
  639:8     warning     Assignment operator must have exactly single space on both sides of it. operator-whitespace
  640:8     warning     Line exceeds the limit of 145 characters                             max-len
  652:9     warning     Only use indent of 12 spaces.                                        indentation
  653:9     warning     Only use indent of 12 spaces.                                        indentation
  681:8     warning     Line exceeds the limit of 145 characters                             max-len
  686:9     warning     Only use indent of 12 spaces.                                        indentation
  689:9     warning     Only use indent of 12 spaces.                                        indentation
  690:9     warning     Only use indent of 12 spaces.                                        indentation
  700:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  719:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  731:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  733:4     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  738:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  743:8     warning     Line contains trailing whitespace                                    no-trailing-whitespace
  744:8     warning     Line exceeds the limit of 145 characters                             max-len
  766:4     warning     Line exceeds the limit of 145 characters                             max-len
  780:8     warning     Line exceeds the limit of 145 characters                             max-len

✖ 166 warnings found.
```