Kwenta Paymaster Audit v1.0.0

Audit by: Jared Borders

Detailed Findings

The following sections include in-depth descriptions of the audit findings.

(C) Critical Risk

> *The issue must be fixed immediately, as significant funds/assets are at risk.*

(H) High Risk

> *The issue must be addressed promptly. Failure to do so could result in the loss of funds/assets or deviation from the provided specifications.*

(M) Medium Risk

> *It is strongly recommended to fix the issue, as its implications could significantly impact the project, though not to an existential extent.*

(L) Low Risk

> *The risk is minor and potentially unlikely or irrelevant, but it is noteworthy.*

(L-1) `percentageMarkup()` Sanitization

Description

If `percentageMarkup` is set to a value less than `100`, the `getCostOfGasInUSDC()` function will return a value strictly less than the actual cost of gas. This could lead to a situation where the user is charged less than the actual cost of gas.

Recommendation

Sanitize the `percentageMarkup` value in the setter to ensure it is always `≥ 100`. Alternatively, use `20` to represent a `20%` markup and adjust the `getCostOfGasInUSDC()` function accordingly.

(Q) Code Quality

> *There is no immediate risk, but fixing the issue would improve code quality, better conform to standards, and potentially reduce unperceived future risks.*

(Q-1) NatSpec: `MarginPaymaster` Immutables

Description

Many of the immutable values in `MarginPaymaster` are not documented.

Recommendation

Add NatSpec describing each immutable/constant value.

(Q-2) Separate `MarginPaymaster`'s Immutables and Constants

Description

The `MarginPaymaster` contract's immutables and constants are placed indiscriminately.

Recommendation

Add a `CONSTANTS` section to the contract and separate immutables from constants.

(Q-3) NatSpec: `MarginPaymaster` Functions

Description

Many functions in `MarginPaymaster` are not documented.

Recommendation

Add NatSpec describing each function, its parameters, and return values.

(Q-4) NatSpec: `MarginPaymaster` Modifier

Description

The `onlyEntryPoint` modifier in `MarginPaymaster` is not documented.

Recommendation

Add NatSpec describing the modifier.

(Q-5) Unused Return Values

Description

The `balance` and `amount` return values in `getUSDCAvailableInWallet()` are never used within the paymaster.

Recommendation

Remove `balance` and `amount` from `getUSDCAvailableInWallet()`.

(Q-6) Duplicate Constant

Description

The `USDC_TO_SUSDC_DECIMALS_INCREASE` constant defined in `MarginPaymaster` is a duplicate of the constant defined in `Zap`.

Recommendation

Use Zap's `_DECIMALS_FACTOR`, which is already defined.

(Q-7) Add Event to `postOp()`

Description

`postOp()` does not emit an event when `withdrawn` is zero.

Recommendation

Add an event to `postOp()` that emits only when `withdrawn` is zero; this will surely help in debugging/monitoring later.

(Q-8) Naked Value in `getCostOfGasInUSDC()`

Description

The value `100` is used in `getCostOfGasInUSDC()` without sufficient context.

Recommendation

Define a constant for `100` in `getCostOfGasInUSDC()` that represents "100%."

(Q-9) Inaccurate Documentation

Description

The documentation in `getWalletAccountId()` states that only one account is supported.

Recommendation

Update the documentation in `getWalletAccountId()` to reflect the actual behavior (i.e., multiple accounts are now supported).

(I) Informational

> *These are warnings and considerations for protocol operation. No immediate action is required.*

(I-1) Inconsistent Formatting

Description

The codebase has inconsistent formatting.

Recommendation

Run `forge fmt`.

(I-2) `TickMath` Library Unused Function

Description

The following function in the `TickMath` library is not used in the project.

Recommendation

Remove `getTickAtSqrtRatio()` from the `TickMath` library.

(I-3) `OracleLibrary` Unused Functions

Description

The following functions in the `OracleLibrary` library are not used in the project.

Recommendation

- Remove `getOldestObservationSecondsAgo()` from `OracleLibrary`.
- Remove `getBlockStartingTickAndLiquidity()` from `OracleLibrary`.
- Remove `getWeightedArithmeticMeanTick()` from `OracleLibrary`.
- Remove `getChainedPrice()` from `OracleLibrary`.

(I-4) `OracleLibrary` Unused Struct

Description

The following struct in the `OracleLibrary` library is not used in the project.

Recommendation

Remove `WeightedTickData` from `OracleLibrary`.

(I-5) `FullMath` Unused Function

Description

The following function in the `FullMath` library is not used in the project.

Recommendation

Remove `mulDivRoundingUp()` from `FullMath`.

(G) Gas Optimizations

> *The suggested optimization would save a significant amount of gas, making it worthwhile to implement despite the development cost.*

(G-1) `getCostOfGasInUSDC()` Early Exit

Description

Potential early exit opportunity in `getCostOfGasInUSDC()` if `withdrawableMargin` is equal to zero.

Recommendation

Update `if (withdrawableMargin < 0) return 0;` to check `<=` for early exit.