F Y E O

Security Code Review Six Sigma Kolme Overview Report

Six Sigma Project

July 2025
Version 1.0

Presented by:
FYEO Inc.
PO Box 147044
Lakewood CO 80214
United States

Security Level
Public

Table of Contents

Executive Summary

Overview

The Six Sigma Project engaged FYEO Inc. to perform a Security Code Review of Six Sigma Kolme.

The assessment was conducted remotely by the FYEO Security Team. Testing took place between April, 2025 - July, 2025, and focused on the following objectives:

- To provide the customer with an assessment of their overall security posture and any risks that were discovered within the environment during the engagement.
- To provide a professional opinion on the maturity, adequacy, and efficiency of the security measures that are in place.
- To identify potential issues and include improvement recommendations based on the results of our tests.

This report summarizes the engagement, tests performed, and findings. It also contains detailed descriptions of the discovered vulnerabilities, steps the FYEO Security Team took to identify and validate each issue, as well as any applicable recommendations for remediation.

Scope and Rules of Engagement

The FYEO Review Team performed a Security Code Review Six Sigma Kolme. The following table documents the targets in scope for the engagement. No additional systems or resources were in scope for this assessment.

The source code was supplied through a private repository at https://github.com/saage-tech/six-sigma-kolme with the commit hash 64666eb420f717fe9808ef8f4f5be881cbc25acc.

Remediations have been submitted with commit hash a99ef6f23523d6366e251b8fdb1a9a68735a6cbd.

Files included in the code review

```
six-sigma-kolme/
├── contracts/
│   ├── orderbook/
│   │   └── src/
│   │       └── lib.rs
│   └── strategic-reserve/
│       └── src/
│           └── lib.rs
├── integration-tests/
│   ├── src/
```

Files included in the code review

```
│   │   ├── api_server_dev_api.rs
│   │   ├── api_server_tasks.rs
│   │   ├── app_tasks.rs
│   │   ├── back_office_tasks.rs
│   │   ├── blockchain_tasks.rs
│   │   ├── broker_fake_data.rs
│   │   ├── broker_tasks.rs
│   │   ├── catalog_tasks.rs
│   │   ├── configs.rs
│   │   ├── dev_apis.rs
│   │   ├── lib.rs
│   │   ├── referral_tasks.rs
│   │   └── setup_tasks.rs
│   └── tests/
│       ├── modules/
│       │   ├── back_office.rs
│       │   ├── bets_warehouse.rs
│       │   ├── email_alerts.rs
│       │   ├── export_data.rs
│       │   ├── favourites.rs
│       │   ├── full_market_cycle.rs
│       │   ├── get_updates.rs
│       │   ├── homepage.rs
│       │   ├── house_provision_outcome_warehouse.rs
│       │   ├── info.rs
│       │   ├── kyc.rs
│       │   ├── leaderboards.rs
│       │   ├── league_management.rs
│       │   ├── login_and_jwt_auth.rs
│       │   ├── mod.rs
│       │   ├── notifications.rs
│       │   ├── provisions_warehouse.rs
│       │   ├── referral.rs
│       │   ├── rewards_backoffice.rs
│       │   ├── search.rs
│       │   ├── settlement_corner_cases.rs
│       │   ├── show_in_explore_corner_cases.rs
│       │   └── usd_exchange_rate.rs
│       └── integration.rs
├── load-tests/
│   └── backoffice-loadtest/
│       └── src/
│           └── main.rs
```

Files included in the code review

```
└── packages/
    ├── admin-cli/
    │   └── src/
    │       ├── leagues.rs
    │       └── main.rs
    ├── api-server/
    │   └── src/
    │       ├── handlers/
    │       │   ├── bets.rs
    │       │   ├── docs.rs
    │       │   ├── favourites.rs
    │       │   ├── fixtures.rs
    │       │   ├── get_updates.rs
    │       │   ├── homepage.rs
    │       │   ├── house_engagements.rs
    │       │   ├── info.rs
    │       │   ├── leaderboard.rs
    │       │   ├── leagues.rs
    │       │   ├── metrics.rs
    │       │   ├── notifications.rs
    │       │   ├── provisions.rs
    │       │   ├── referrals.rs
    │       │   ├── social_sharing.rs
    │       │   ├── usd_exchange_rates.rs
    │       │   └── users.rs
    │       ├── models/
    │       │   ├── bet.rs
    │       │   ├── favourite.rs
    │       │   ├── fixture.rs
    │       │   ├── homepage.rs
    │       │   ├── house_engagement.rs
    │       │   ├── leaderboard.rs
    │       │   ├── market.rs
    │       │   ├── notification.rs
    │       │   ├── provision.rs
    │       │   ├── referrals.rs
    │       │   └── updates.rs
    │       ├── api_error.rs
    │       ├── api_json.rs
    │       ├── app.rs
    │       ├── authorization_middleware.rs
    │       ├── bin.rs
    │       ├── common.rs
```

Files included in the code review

```
│       ├── debug.rs
│       ├── handlers.rs
│       ├── lib.rs
│       ├── models.rs
│       ├── routes.rs
│       ├── settings.rs
│       └── top_market_loader.rs
├── axum-jsonschema/
│   └── src/
│       └── lib.rs
├── back-office/
│   ├── back-office-web/
│   │   ├── src/
│   │   │   ├── types/
│   │   │   │   └── index.ts
│   │   │   ├── utils/
│   │   │   │   ├── balances/
│   │   │   │   │   ├── interface.ts
│   │   │   │   │   ├── useBettingBalances.ts
│   │   │   │   │   └── useWalletBalances.ts
│   │   │   │   ├── kolme/
│   │   │   │   │   ├── helper.ts
│   │   │   │   │   ├── useChainConfig.ts
│   │   │   │   │   └── useKolmeTx.ts
│   │   │   │   ├── solana/
│   │   │   │   │   ├── interface.ts
│   │   │   │   │   ├── pda.ts
│   │   │   │   │   ├── spl.ts
│   │   │   │   │   ├── useDataExtractors.ts
│   │   │   │   │   ├── useQuery.ts
│   │   │   │   │   └── useSolanaBridgeTx.ts
│   │   │   │   ├── string/
│   │   │   │   │   ├── index.ts
│   │   │   │   │   └── string.ts
│   │   │   │   ├── auth.ts
│   │   │   │   ├── bigdecimal.ts
│   │   │   │   ├── chainApi.ts
│   │   │   │   ├── crypto.ts
│   │   │   │   ├── decode.ts
│   │   │   │   ├── fetchJson.ts
│   │   │   │   ├── token.ts
│   │   │   │   ├── tx.ts
│   │   │   │   ├── utils.ts
```

Files included in the code review

```
│   │   │   │   └── websockets.ts
│   │   │   └── vite-env.d.ts
│   │   └── vite.config.ts
│   └── src/
│       ├── handlers/
│       │   ├── accounting.rs
│       │   ├── address_group.rs
│       │   ├── backoffice_users.rs
│       │   ├── bets_warehouses.rs
│       │   ├── business_analytics.rs
│       │   ├── campaigns.rs
│       │   ├── config.rs
│       │   ├── countries.rs
│       │   ├── docs.rs
│       │   ├── emergency.rs
│       │   ├── fixtures.rs
│       │   ├── house_provision_by_address.rs
│       │   ├── house_provision_by_fixture.rs
│       │   ├── house_provision_outcome_warehouses.rs
│       │   ├── house_provision_warehouses.rs
│       │   ├── leagues.rs
│       │   ├── live_dashboard.rs
│       │   ├── market_types.rs
│       │   ├── market_types_outcomes.rs
│       │   ├── markets.rs
│       │   ├── metrics.rs
│       │   ├── profit_warehouses.rs
│       │   ├── referrals.rs
│       │   ├── reward.rs
│       │   ├── teams.rs
│       │   ├── usd_exchange_rate.rs
│       │   └── user_warehouses.rs
│       ├── models/
│       │   ├── accounting.rs
│       │   ├── address_group.rs
│       │   ├── backoffice_user.rs
│       │   ├── bet_warehouse.rs
│       │   ├── business_analytics.rs
│       │   ├── campaigns.rs
│       │   ├── fixture.rs
│       │   ├── house_provision_outcome_warehouse.rs
│       │   ├── house_provision_warehouse.rs
│       │   ├── league.rs
```

Files included in the code review

```
│       │   ├── live_dashboard.rs
│       │   ├── market.rs
│       │   ├── profit_warehouse.rs
│       │   ├── referrals.rs
│       │   ├── reward.rs
│       │   ├── team.rs
│       │   ├── usd_exchange_rate.rs
│       │   └── user_warehouse.rs
│       ├── api_error.rs
│       ├── api_json.rs
│       ├── app.rs
│       ├── authorization_middleware.rs
│       ├── bin.rs
│       ├── handlers.rs
│       ├── lib.rs
│       ├── models.rs
│       └── settings.rs
├── bots/
│   └── src/
│       ├── endpoints/
│       │   ├── common.rs
│       │   └── status.rs
│       ├── app.rs
│       ├── balance_check.rs
│       ├── crank.rs
│       ├── endpoints.rs
│       ├── exchange_rates.rs
│       ├── main.rs
│       ├── metrics.rs
│       └── settings.rs
├── catalog-and-users-service/
│   └── src/
│       ├── bin/
│       │   └── catalog-and-users-service.rs
│       ├── indexer/
│       │   ├── db/
│       │   │   ├── bank.rs
│       │   │   ├── bets.rs
│       │   │   ├── house_provision.rs
│       │   │   ├── market.rs
│       │   │   ├── market_parameters.rs
│       │   │   ├── process.rs
│       │   │   ├── reward.rs
```

Files included in the code review

```
│   │   │   ├── tx.rs
│   │   │   └── types.rs
│   │   ├── db.rs
│   │   ├── processor.rs
│   │   └── six_sigma_event.rs
│   ├── models/
│   │   ├── accounting.rs
│   │   ├── address_group.rs
│   │   ├── backoffice_user.rs
│   │   ├── bet_warehouse.rs
│   │   ├── business_analytics.rs
│   │   ├── campaign.rs
│   │   ├── checkpoint.rs
│   │   ├── engagement_warehouse.rs
│   │   ├── favourite.rs
│   │   ├── fixture.rs
│   │   ├── house_provision_outcome_warehouse.rs
│   │   ├── house_provision_stats.rs
│   │   ├── house_provision_warehouse.rs
│   │   ├── leaderboard.rs
│   │   ├── league.rs
│   │   ├── market.rs
│   │   ├── notification.rs
│   │   ├── profit_warehouse.rs
│   │   ├── referral.rs
│   │   ├── reward.rs
│   │   ├── team.rs
│   │   ├── usd_exchange_rate.rs
│   │   ├── user_account.rs
│   │   ├── user_wallet_seed.rs
│   │   └── user_warehouse.rs
│   ├── app.rs
│   ├── config.rs
│   ├── error.rs
│   ├── export.rs
│   ├── healthz.rs
│   ├── indexer.rs
│   ├── kyc.rs
│   ├── lib.rs
│   ├── logging.rs
│   ├── metrics.rs
│   ├── models.rs
│   ├── pager_duty.rs
```

Files included in the code review

```
│       ├── templates.rs
│       ├── test_helpers.rs
│       └── wallet.rs
├── countries/
│   └── src/
│       └── lib.rs
├── deposit-bonus-token/
│   └── src/
│       └── lib.rs
├── kolme-app/
│   └── src/
│       ├── api_server/
│       │   ├── client.rs
│       │   └── error.rs
│       ├── app/
│       │   └── state.rs
│       ├── api_server.rs
│       ├── app.rs
│       ├── client.rs
│       ├── lib.rs
│       ├── main.rs
│       └── metrics.rs
├── kyc-token/
│   └── src/
│       └── lib.rs
├── lsports-broker/
│   └── src/
│       ├── db/
│       │   └── chain_sender.rs
│       ├── lsports/
│       │   ├── amqp.rs
│       │   ├── leagues_api.rs
│       │   ├── snapshot_api.rs
│       │   └── types.rs
│       ├── processor/
│       │   ├── chain_sender.rs
│       │   ├── league_poller.rs
│       │   └── waits.rs
│       ├── bin.rs
│       ├── cli.rs
│       ├── client.rs
│       ├── db.rs
│       ├── db_migrator.rs
```

Files included in the code review

```
│       ├── healthz.rs
│       ├── ingester.rs
│       ├── lib.rs
│       ├── lsports.rs
│       ├── macros.rs
│       ├── metrics.rs
│       ├── odds_server.rs
│       ├── processor.rs
│       ├── test_helpers.rs
│       ├── tests.rs
│       ├── types.rs
│       └── util.rs
├── market-types/
│   └── src/
│       └── lib.rs
├── orderbook-lib/
│   └── src/
│       ├── orderbook/
│       │   ├── state/
│       │   │   ├── account.rs
│       │   │   ├── admin.rs
│       │   │   ├── bet.rs
│       │   │   ├── market.rs
│       │   │   ├── mod.rs
│       │   │   ├── prefix_sum.rs
│       │   │   ├── rewards.rs
│       │   │   └── sparse_queue.rs
│       │   ├── fnv.rs
│       │   ├── mod.rs
│       │   ├── range.rs
│       │   ├── response.rs
│       │   └── token.rs
│       ├── tests/
│       │   └── serializing.rs
│       ├── lib.rs
│       └── tests.rs
├── orderbook-research/
│   └── src/
│       ├── lib.rs
│       ├── orderbook.rs
│       ├── orderbook_printer.rs
│       ├── simulate.rs
│       ├── tests.rs
```

Files included in the code review
│ ├── types.rs │ └── utils.rs ├── secret-fetch/ │ └── src/ │ └── lib.rs └── solana-client/ └── src/ └── lib.rs

Table 1: Scope

Technical Analyses and Findings

During the Security Code Review Six Sigma Kolme, we discovered:

- 2 findings with MEDIUM severity rating.
- 5 findings with LOW severity rating.
- 13 findings with INFORMATIONAL severity rating.

The following chart displays the findings by severity.

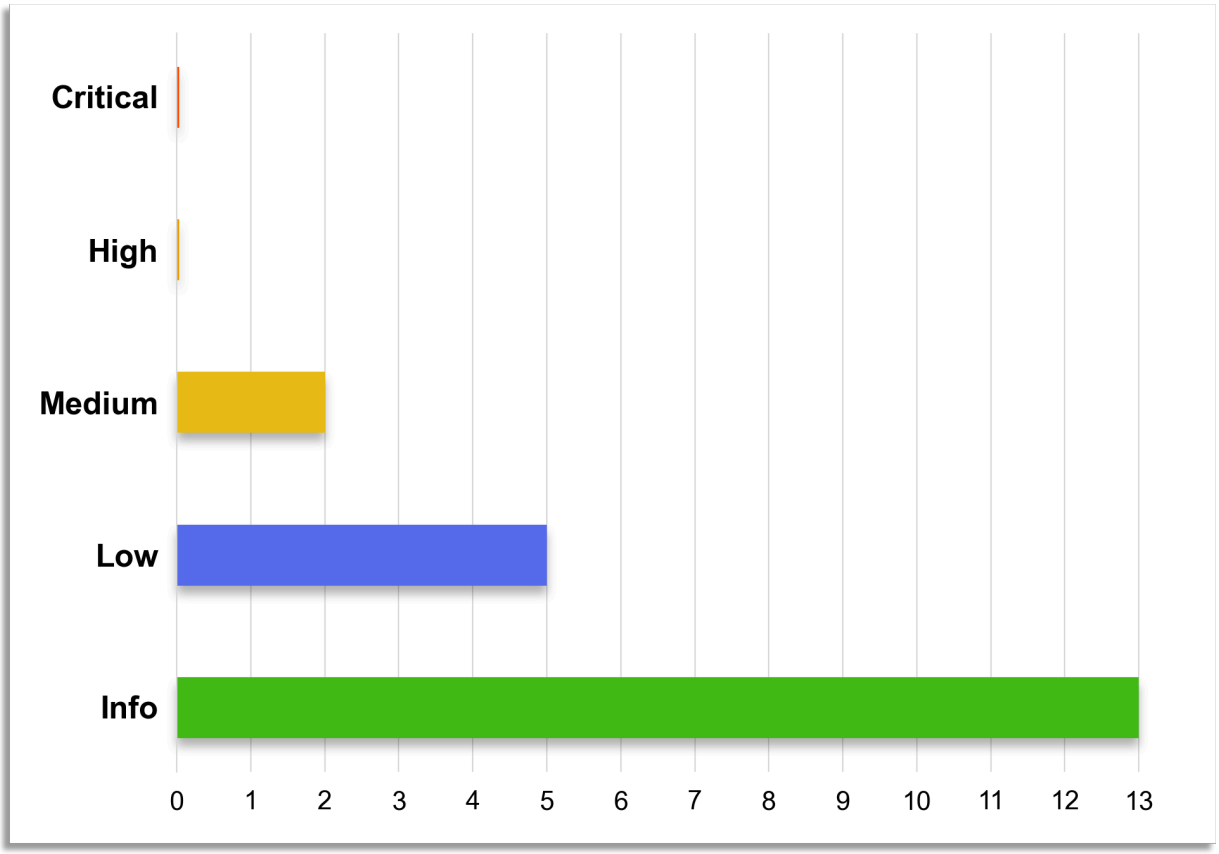

Figure 1: Findings by Severity

The Classification of vulnerabilities

Security vulnerabilities and areas for improvement are weighted into one of several categories using, but is not limited to, the criteria listed below:

Critical – vulnerability will lead to a loss of protected assets

- This is a vulnerability that would lead to immediate loss of protected assets
- The complexity to exploit is low
- The probability of exploit is high

High - vulnerability has potential to lead to a loss of protected assets

- All discrepancies found where there is a security claim made in the documentation that cannot be found in the code
- All mismatches from the stated and actual functionality
- Unprotected key material
- Weak encryption of keys
- Badly generated key materials
- Txn signatures not verified
- Spending of funds through logic errors
- Calculation errors overflows and underflows

Medium - vulnerability hampers the uptime of the system or can lead to other problems

- Insecure calls to third party libraries
- Use of untested or nonstandard or non-peer-reviewed crypto functions
- Program crashes, leaves core dumps or writes sensitive data to log files

Low – vulnerability has a security impact but does not directly affect the protected assets

- Overly complex functions
- Unchecked return values from 3rd party libraries that could alter the execution flow

Informational

- General recommendations

Technical Analysis

The source code has been manually validated to the extent that the state of the repository allowed. The validation includes confirming that the code correctly implements the intended functionality.

Conclusion

Based on our review process, we conclude that the code implements the documented functionality to the extent of the reviewed code.

Technical Findings

General Observations

The Six Sigma App has evolved into a comprehensive command-line interface built atop the specialized Kolme blockchain architecture, enabling a full sports-betting lifecycle from market creation through settlement. The latest refactor explicitly manages and serializes all state transitions into Kolme-compatible data structures, while smart contracts have been cleanly factored into their own crate alongside a lightweight client and a shared orderbook-lib crate. The backend ecosystem now comprises dedicated Rust crates—admin-cli, api-server, back-office, bots, Six Sigma-app, lsports-broker, and more—plus domain-specific companion crates (kyc-token, deposit-bonus-token, countries), all supported by a robust integration-tests harness and a load-tests binary that validate cross-chain synchronization and critical back-office endpoints.

On the API and data-model front, error handling has been unified and hardened: responses are standardized, sensitive details are concealed from clients, and all HTTP routes now include schema validation, CORS support, health checks, and request tracing for full observability. Numeric types have been consolidated under rust_decimal::Decimal, integer fields standardized, and unsafe subtraction behaviors replaced with underflow-guard helpers. Identifier schemes shifted from address-based contracts to numeric account IDs, composite keys enforce data integrity, and database migrations reflect these domain renames. New administrative controls and referral-reward flows have been introduced, markets can be capped and seeded, and configuration now supports multi-key setups, P2P networking, and dynamic initialization. Client and ingestion components leverage asynchronous subscriptions and concurrency primitives to stay synchronized with on-chain events. Front-end support has grown with a USD exchange-rate endpoint, multi-currency market types, secret storage for account updates, and React hooks that fetch on-chain and wallet balances, handle decimal precision, load blockchain configs, and perform unified query operations. Solana integrations now include PDAs, automatic token-account creation, and a WebSocket bridge stream with caching, promise-based lookups, auto-reconnect, and robust error checking, all underpinned by utility functions for formatting, unit conversions, and light mismatch warnings.

Across all diffs, security reviews uncovered and addressed numerous vulnerabilities: server binding to any interface, raw secrets exposed on command lines, insufficient HTTP timeouts, denial-of-service vectors, unwrapping panics, stale balance checks, and potential resource exhaustion from large payloads. Code-level issues included debug formatting that could leak internal state, unparameterized SQL ILIKE clauses risking injection, unauthenticated Prometheus endpoints, weak client-side encryption employing SHA-256 instead of a proper KDF, signing keys stored in the localStorage, hardcoded test keys in source, Solana API keys exposed, and token-balance helpers that fail to verify account ownership. Remediations have introduced PBKDF2 with salt and IV for key derivation, underflow guards, standardized error-hiding, query parameterization, authentication on metrics endpoints, and overall tightening of secret handling and encryption practices.

Our Process

Methodology

FYEO Inc. uses the following high-level methodology when approaching engagements. They are broken up into the following phases.

Figure 2: Methodology Flow

Kickoff

The project is kicked off as the sales process has concluded. We typically set up a kickoff meeting where project stakeholders are gathered to discuss the project as well as the responsibilities of participants. During this meeting we verify the scope of the engagement and discuss the project activities. It's an opportunity for both sides to ask questions and get to know each other. By the end of the kickoff there is an understanding of the following:

- Designated points of contact
- Communication methods and frequency
- Shared documentation
- Code and/or any other artifacts necessary for project success
- Follow-up meeting schedule, such as a technical walkthrough
- Understanding of timeline and duration

Ramp-up

Ramp-up consists of the activities necessary to gain proficiency on the project. This can include the steps needed for familiarity with the codebase or technological innovation utilized. This may include, but is not limited to:

- Reviewing previous work in the area including academic papers
- Reviewing programming language constructs for specific languages
- Researching common flaws and recent technological advancements

Review

The review phase is where most of the work on the engagement is completed. This is the phase where we analyze the project for flaws and issues that impact the security posture. Depending on the project this may include an analysis of the architecture, a review of the code, and a specification matching to match the architecture to the implemented code.

In this code audit, we performed the following tasks:

1. Security analysis and architecture review of the original protocol
2. Review of the code written for the project
3. Compliance of the code with the provided technical documentation

The review for this project was performed using manual methods and utilizing the experience of the reviewer. No dynamic testing was performed, only the use of custom-built scripts and tools were used to assist the reviewer during the testing. We discuss our methodology in more detail in the following sections.

Code Safety

We analyzed the provided code, checking for issues related to the following categories:

- General code safety and susceptibility to known issues
- Poor coding practices and unsafe behavior
- Leakage of secrets or other sensitive data through memory mismanagement
- Susceptibility to misuse and system errors
- Error management and logging

This list is general and not comprehensive, meant only to give an understanding of the issues we are looking for.

Technical Specification Matching

We analyzed the provided documentation and checked that the code matches the specification. We checked for things such as:

- Proper implementation of the documented protocol phases
- Proper error handling
- Adherence to the protocol logical description

Reporting

FYEO Inc. delivers a draft report that contains an executive summary, technical details, and observations about the project.

The executive summary contains an overview of the engagement including the number of findings as well as a statement about our general risk assessment of the project. We may conclude that the overall risk is low but depending on what was assessed we may conclude that more scrutiny of the project is needed.

We report security issues identified, as well as informational findings for improvement, categorized by the following labels:

- Critical
- High
- Medium
- Low
- Informational

The technical details are aimed more at developers, describing the issues, the severity ranking and recommendations for mitigation.

As we perform the audit, we may identify issues that aren't security related, but are general best practices and steps that can be taken to lower the attack surface of the project. We will call those out as we encounter them and as time permits.

As an optional step, we can agree on the creation of a public report that can be shared and distributed with a larger audience.

Verify

After the preliminary findings have been delivered, this could be in the form of the approved communication channel or delivery of the draft report, we will verify any fixes within a window of time specified in the project. After the fixes have been verified, we will change the status of the finding in the report from open to remediated.

The output of this phase will be a final report with any mitigated findings noted.

Additional Note

It is important to note that, although we did our best in our analysis, no code audit or assessment is a guarantee of the absence of flaws. Our effort was constrained by resource and time limits along with the scope of the agreement.

While assessing the severity of the findings, we considered the impact, ease of exploitability, and the probability of attack. This is a solid baseline for severity determination.