**Before diving into the deep end of code analysis, a well-defined scope and thorough reconnaissance are crucial for a successful smart contract audit.**

**Scoping:**

- **Defining the Boundaries:**
    - We will collaboratively establish the precise aspects of your smart contract to be audited.
    - This might encompass all functionalities within the code, or it could be focused on specific modules or features based on your priorities.
- **Aligning Expectations:**
    - A clear scope ensures both parties understand what's included and excluded from the audit.
    - This helps manage project timelines and costs effectively.
- **Focus on Critical Areas:**
    - By delineating the scope, we can prioritize critical functionalities and potential risks for a more targeted and efficient audit.

**Reconnaissance:**

- **Gathering Information:**
    - This initial phase involves collecting all relevant documentation and context about your smart contract.
    - This includes the items listed in the previous section (NatSpec, Dependencies, etc.) as well as additional resources like project whitepapers and roadmaps.
- **Understanding the System:**
    - Through code review and analysis, we gain a comprehensive understanding of how your smart contract interacts with other components within the system (dApps, other contracts, etc.).
- **Identifying Potential Risks:**
    - During reconnaissance, we can flag potential vulnerabilities based on common attack vectors and best practices.
    - This early identification allows for a more focused audit and helps prioritize areas of concern.

| Severity | Description |
|---|---|
| CRITICAL | client must fix the issue, no matter what, because not fixing will lead to significant funds/assets WILL loss. |
| HIGH | client must address the issue, no matter what, because not fixing would be very bad, or some funds/assets will be lost |
| MEDIUM | the client to seriously consider fixing the issue, as the implications of not fixing the issue are severe enough to impact the project significantly |
| LOW | Low The risk is small, unlikely, or may not be relevant to the project in a meaningful way. |
| Code Quality | issue identified does not pose any obvious risk, but fixing could improve overall code quality |
| INFORMATIONAL | Informational Warnings and things to keep in mind when operating the protocol |

Audit findings are  based on their potential impact on your smart contract.

To ensure a thorough and efficient smart contract audit, I kindly request you to provide the following information:

**1. Clearly Defined Natural Language Specification (NatSpec):**

- A human-readable document outlining the intended functionalities and behaviour of your smart contract.
- This document should be clear, concise, and easy to understand, even for those without technical expertise.
- The NatSpec serves as a common ground to ensure both parties (you and I) have the same understanding of the smart contract's purpose.

**2. Dependencies:**

- A detailed list of all external libraries, frameworks, or other smart contracts your smart contract relies upon.
- Understanding these dependencies is crucial for assessing potential vulnerabilities that might exist within them and how they interact with your code.

**3. Interfaces:**

- Provide a clear description of how your smart contract interacts with external systems or users.
- This includes function parameters, return values, and any relevant error codes.

- A well-defined interface ensures smooth integration and reduces the risk of unexpected behaviour.

**4. Detailed State Changes:**

- A comprehensive explanation of how the state of your smart contract (data it stores) is modified during function calls.
- This includes initial states, valid transitions, and expected outcomes.
- Understanding state changes is essential for verifying data integrity and preventing unintended modifications.

**5. Events:**

- A thorough breakdown of all events emitted by your smart contract.
- Events are notifications that signal specific occurrences within the contract's execution.
- Understanding events allows for monitoring and integration with external systems.

**6. Exception Handling:**

- A clear explanation of how your smart contract handles unexpected situations or errors.
- This includes how errors are identified, reported, and potentially rolled back if necessary.
- Robust exception handling prevents the contract from getting stuck in unforeseen states.

**7. Actors and Access Controls:**

- A definition of all the different roles (actors) that can interact with your smart contract.
- Specify the permissions assigned to each actor, outlining what functions they can call and what data they can access.
- Granular access control minimizes the attack surface and protects sensitive information.

By providing this information upfront, you will significantly enhance the efficiency and effectiveness of your smart contract audit. It allows for a clear understanding of your project's goals and facilitates a more focused analysis. If you have any questions or require further clarification on these points, please don't hesitate to ask.

**Benefits of a Structured Scoping and Reconnaissance Phase:**

- **Enhanced Efficiency:**
  - By defining the scope and conducting thorough reconnaissance, we can streamline the audit process, focusing resources on areas of highest risk.
- **Reduced Costs:**
  - A clear scope prevents scope creep (expanding the audit beyond initial agreements) which can lead to unexpected cost increases.
- **Improved Communication:**

    - Open communication during scoping and reconnaissance fosters a collaborative environment and ensures everyone is aligned on priorities.

**Moving Forward:**

Once scoping and reconnaissance are complete, we can move on to the in-depth analysis phase of the audit. With a solid understanding of your project and its goals, I can deliver a comprehensive and targeted audit that maximizes the security and reliability of your smart contract.