

Audit Checklist

Ethereum Smart Contracts – Version 1.3

Send us the scope for the audit:

- ☐ Send us relevant documentation such as whitepapers, websites or online documentation
- ☐ Send us the repositories and commit hash that should be used for the quote
- ☐ Specify the relevant directories/files in the repositories for the audit
- ☐ Send us an outline of how the scope/code will change before the audit begins, including a detailed description of any still unimplemented parts

In addition, before the audit start:

- ☐ Resolve any compiler warnings
- ☐ Run a linter, for example [solhint](#)
- ☐ Ensure all tests of your projects pass
- ☐ Ensure you have high test coverage. To start an audit, we require at least 70%, ideally 90% test coverage. You can compute test coverage, e.g., through [soldity-coverage](#):
`npx hardhat coverage`
- ☐ Remove any files that are not specific to your project, e. g. Markdown files from a template you used
- ☐ Remove any commented/unused code
- ☐ Resolve any pending TODOs
- ☐ Send us the repositories and commit hash that should be used for the audit (code freeze)
- ☐ Specify the relevant directories/files in the repositories for the audit
- ☐ Make sure that your project builds correctly on clean system by pulling all the necessary dependencies
- ☐ Document your build and test system (e.g. add an example `.env` file for local deploys)
- ☐ Send us the computed test coverage