Quick start (developer)

These are generic Haskell/Plutus steps. Adapt to your local toolchain (Nix, Cabal, Stack) and to the project's CI.

1. **Prerequisites**
    - Haskell toolchain (GHC + Cabal or Stack) and Cabal/Stack on PATH
    - Nix (optional) if your project uses Nix for reproducible builds
    - Plutus libraries and dependencies as defined in the project `cabal.project/package.yaml`
2. **Clone the repository**

```
git clone https://github.com/your-org/your-repo.git
cd your-repo/SmartContract
```

3. **Build** (Cabal)

```
cabal build
```

Or with Stack:

```
stack build
```

If using Nix, follow the repo's `README` for `nix-shell` and build instructions.

4. **Run tests**

- Unit tests and property tests (Tasty + QuickCheck) are typically run via:

```
cabal test
# or
stack test
```

5. **Generate `.plutus` files / compile validators**

- Many Plutus projects expose a `plutus` or `scripts` target to produce serialized validator files. Check the repo's build scripts or `Makefile`.

6. **Local REPL / debugging**

```
cabal repl
# then import modules: :l PCMint
```

Testing & CI recommendations

- Keep QuickCheck-based property tests (as you have been doing) for critical invariants: signature checks, datum format, duplicate ID checks, reputation calculations, role grants, and registry updates.
- Use Tasty to group unit tests and to run deterministic test cases for both happy and failure paths.

- Add a CI job to compile all modules and run tests on pushes and PRs.

Security & deployment notes

- Keep private keys, signing keys, and KMS secrets out of source control. Use Google Secret Manager / AWS Secrets Manager / HashiCorp Vault.
- Validate datums and redeemers strictly on-chain; prefer inline datums for state flows where appropriate.
- Use a staging/testnet flow and automated integration tests before any mainnet deploy.

Contributing

1. Fork the repo and create a feature branch.
2. Run tests locally and ensure new code is covered by tests.
3. Open a PR with a clear description and link to any design notes.