ANASTASIA LABS

Proof of Achievement - Milestone 1

OpShin Language Analysis Report

Project Number 1200175
Project Manager Jonathan Rodriguez

Contents

Quantitative Metrics **1**
Introduction 1
Key Features 1
Limitations 1
Code Coverage Percentage **2**
Number of Manual Review Findings – divide this topic into categories **3**
Recommendations for Improvements 4

Project Name: OpShin Audit
URL: Catalyst Proposal

Quantitative Metrics

Introduction

OpShin is an pythonic smart contract language designed for the Cardano blockchain ecosystem. It aims to lower the barrier to entry for Cardano smart contract development by leveraging Python's syntax and ecosystem while compiling to Plutus Core. This approach allows developers to write smart contracts using familiar Python constructs while still benefiting from Cardano's robust and secure execution environment.

Key Features

Limitations

1. Its primary support is for List and Dict types,Lack of native tuple support introduces workarounds.

Code Coverage Percentage

Number of Manual Review Findings – divide this topic into categories

1. **Unclear Error Messages**:

When executing OpShin scripts using the command `opshin eval any filepath`, the error messages produced are not sufficiently clear or informative. This lack of clarity in error reporting can hinder developers' ability to quickly identify and resolve issues in their smart contract code.

2. **Static Type Inference Testing**: under Security

The static type inference system in OpShin lacks comprehensive testing, particularly for scenarios involving dynamically passed variables. There is a need to develop and implement additional test cases to verify how the static type system infers types for variables that are passed dynamically during runtime. This testing gap could potentially lead to unexpected behavior or type-related errors in smart contracts.

3. **Compiled Code**:

The compiled code for Minting and Spending purpose seems to be different with respect to the force-three-params tag.

4. **UI(Some other wors) for `blueprint.json file`** :

The json file does not clearly state the title of the validator name, We can recommend the opshin team to add the title of the validator which can be used as aredernece while writing off chain code.

5. No support for **tuple** and **generic types**

6. To be discussed —> (more complex unit test cases) -> Milestone4

Recommendations for Improvements