Decentralized Inheritance Protocol

# Report

Noah Klaholz
Department / Organization

A short project report for the Decentralized Inheritance Protocol implementation.
Refer to the repository for code and tests.

November 5, 2025

# Contents

**1 Introduction** **2**

**2 Project Overview** **2**

**3 Repository Structure** **2**

**4 Usage and Build** **2**

**5 Appendix** **2**

**Appendices** **2**

**A Sample Code Listing** **2**

**B References** **3**

## 1 Introduction

This document contains a short report for the Decentralized Inheritance Protocol project. It provides an overview of the project, features implemented, and references to the codebase. The project README contains a roadmap and design notes which are reflected here.

## 2 Project Overview

Based on the repository README, the project aims to implement a smart-contract-based inheritance protocol with the following implemented features:

- Beneficiary management logic (implemented and tested)
- Transferring and withdrawing assets using a mock USDC token (implemented and tested)
- State handling via check-ins to reset a liveness timer (implemented and tested)
- Phase transitions (Active → Warning → Verification → Distribution) (implemented and tested)
- Verification phase with mocked death certificates/death oracle (implemented and tested)
- Payout logic with percentage-based shares (implemented and tested)

## 3 Repository Structure

The repository contains a ‘contract/‘ folder with Hardhat configuration, Solidity contracts, tests and deployment artifacts, and a ‘client/‘ folder for potential frontend code. The ‘docs/‘ folder is used for documentation; this report lives in ‘docs/report‘.

## 4 Usage and Build

The recommended cross-platform build uses the provided scripts in this folder. The scripts run ‘latexmk‘ where available or fall back to ‘pdflatex‘ + ‘biber‘ as necessary. The build process writes intermediary files to ‘docs/report/build/‘ which is ignored by git; only the resulting ‘report.pdf‘ and source files are kept in version control.

## 5 Appendix

Refer to the appendix for extra materials.

# Appendices

## A Sample Code Listing

Below is an example of how to include a code listing using the ‘listings‘ package. Replace with actual code when needed.

Listing 1: Example JS snippet

```
// Example: simple function
function greet(name) {
  return `Hello, ${name}`;
```

```
}
console.log(greet('World'));
```

## B References