

# Inheritor - On-Chain Digital Inheritance

---

## About Inheritor

Inheritor is a blockchain-based solution for creating digital wills that securely transfer digital assets to designated beneficiaries upon the testator's incapacitation or death. This repository contains the emergency management tools designed to serve as failsafe mechanisms, ensuring that digital inheritances remain accessible even without the Inheritor mobile application.

## Repository Contents

```
UserRecovery/
├── Inheritor - White Paper.pdf
├── Manuals/
│   ├── CheckClaimable.md
│   ├── CheckClaimable.pdf
│   ├── ClaimManual.md
│   ├── ClaimManual.pdf
│   ├── TestatorManual.md
│   └── TestatorManual.pdf
├── Mitigation.pdf
├── README.md
├── README.pdf
├── faq/
│   ├── faq.md
│   └── faq.pdf
├── mitigation.md
├── package-lock.json
├── package.json
└── scripts/
    ├── Beneficiary_CheckClaimable.js
    ├── Beneficiary_Claim.js
    └── Testator.js
```

## Emergency Tool Suite

This repository provides three essential command-line tools for managing digital inheritances in emergency situations:

### 1. Testator Emergency Management Tool ([scripts/Testator.js](#))

For testators (will creators) to manage their Digital Will when the Inheritor app is unavailable:

- View the status of all inheritances in your Digital Will
- Perform check-ins to reset inheritance timers
- Revoke inheritances to permanently cancel them
- Remove verifiers from your Digital Will

- Return unused ETH to your gas wallet

## 2. Beneficiary Check Tool ([scripts/Beneficiary\\_CheckClaimable.js](#))

For beneficiaries to monitor and check their incoming inheritances:

- View all inheritances designated to your address
- Check if specific inheritances are claimable (and potentially trigger state changes)
- Refund unused ETH to your gas wallet

## 3. Beneficiary Claim Tool ([scripts/Beneficiary\\_Claim.js](#))

For beneficiaries to claim and decrypt their inherited digital assets:

- Verify if an inheritance is in the "Claimable" state
- Retrieve the Arweave transaction ID from the blockchain
- Fetch and decrypt the symmetric key
- Download and decrypt the inherited digital asset
- Save the decrypted file to your local system

## Key Features

- **Self-Sovereign Solution:** No dependence on Inheritor's infrastructure or the mobile app
- **Blockchain-Based:** Leverages the security and immutability of Ethereum and Arbitrum blockchains
- **Cryptographically Secure:** Uses robust encryption to protect sensitive digital assets
- **Permanently Stored:** Digital assets are stored on Arweave's permanent storage network
- **User-Controlled:** Complete control over your digital inheritance process

## Getting Started

Each tool has detailed installation and usage instructions in its corresponding manual:

- Testator Tool: See [Manuals/TestatorManual.md](#)
- Beneficiary Check Tool: See [Manuals/CheckClaimable.md](#)
- Beneficiary Claim Tool: See [Manuals/ClaimManual.md](#)

## Prerequisites

- Node.js (v16.0.0 or higher)
- Basic command-line knowledge
- For testators: Your testator recovery mnemonic and a gas wallet with ETH
- For beneficiaries: Your beneficiary recovery mnemonic and a gas wallet with ETH

## Documentation

- **White Paper:** [Inheritor - White Paper.pdf](#) - Comprehensive overview of the Inheritor system architecture and cryptographic design
- **FAQ:** [faq/faq.md](#) - Frequently asked questions about Inheritor

- **Mitigation Strategy:** [mitigation.md](#) - How Inheritor ensures your digital will remains accessible regardless of the company's existence

## Security Considerations

These tools provide direct access to your blockchain assets and sensitive cryptographic keys. Always follow these security practices:

- Run these tools only on secure, trusted devices
- Never share your recovery mnemonic or private keys
- Use these tools only in genuine emergency situations when the Inheritor app is unavailable
- Verify all transaction details before confirming
- Move any claimed assets to secure storage immediately

## Technical Overview

Inheritor uses a combination of technologies to provide secure digital inheritance:

- **Ethereum/Arbitrum:** Smart contracts that define and enforce inheritance conditions
- **Arweave:** Permanent, decentralized storage for encrypted digital assets
- **Public-key Cryptography:** ECDH key exchange and HKDF for secure key derivation
- **Symmetric Encryption:** AES-GCM for asset encryption and decryption

The system creates a cryptographically secure time-lock mechanism where assets can only be accessed by beneficiaries when specific conditions (like testator inactivity) are met.

## Installation

To install the required dependencies:

```
npm install
```

This will install all the Node.js packages needed to run the scripts, as specified in package.json.

## License

This project is released into the public domain. See [LICENSE](#) for more details.

## Contributing

Contributions to improve these emergency tools are welcome. Please feel free to submit issues or pull requests.

## Contact

For support or questions, contact: [support@inheritor.app](mailto:support@inheritor.app)

**Disclaimer:** These tools are provided for emergency use only. While efforts have been made to ensure their security and accuracy, use them at your own risk. Always verify the effects of any blockchain transactions, as they cannot be reversed once confirmed.

## Acknowledgments

- The Ethereum and Arbitrum communities
- The Arweave project for permanent storage solutions
- All contributors to the Node.js packages used in this project