

BALT

Whitepaper
Bitcoin Autonomous Legacy Trust

Table of Contents

1. Introduction.....	2
2. Vision	2
3. Solution architecture.....	2
3.1 Inheritance Vault.....	2
3.2 Inheritance Registry	2
3.3 Check-in.....	2
3.4 Claim.....	2
3.5 Cancellation.....	2
4. Usage Flow	2
5. Security and Cryptographic Assumptions	3
6. Limitations and Risks	3
7. Roadmap	3
8. Legal Disclaimer.....	3
9. Equipment.....	3
9.1 Contact:	3

1. Introduction

Bitcoin has transformed the concept of financial sovereignty, allowing individuals to custody their own wealth without intermediaries. However, this autonomy also presents a crucial challenge: what happens to the funds when their owner dies or loses access?

BALT (Bitcoin Autonomous Legacy Trust) is a decentralized application built on top of **Rootstock (RSK)**, which allows you to schedule a secure and non-custodial inheritance from third parties, transferring assets to a designated heir in the event of prolonged inactivity.

2. Vision

Our vision is to extend Bitcoin's durability and trust beyond human lifetime, ensuring that accumulated value can be transmitted in a programmed, verifiable, and decentralized manner.

"Make Bitcoin unstoppable — even beyond our lifetime."

3. Solution architecture

BALT operates using smart contracts deployed on the Rootstock Testnet (and soon Mainnet). The architecture is based on the following components:

3.1 Inheritance Vault

- Smart contract deployed by the creator.
- It defines a "downtime" measured in months (seconds for Beta Test Flow).

3.2 Inheritance Registry

- The creator deposits an amount of RBTC into the Vault.
- The heir's address is recorded.
- A commission of 0.5% of the deposited amount is applied, which is automatically deducted at the time of registration of the inheritance.

3.3 Check-in

- The creator can run a "Check-in" at any time.
- This resets the idle counter, preventing the activation of the inheritance.

3.4 Claim

- If the period of inactivity is fulfilled and there were no check-ins, the heir can claim the inheritance from their wallet through the contract.

3.5 Cancellation

- The creator may cancel the inheritance at any time prior to its release. By doing so, the deposited amount is automatically returned to the creator's original wallet.

4. Usage Flow

1. Wallet connection and RSK Testnet network selection.
2. Creating the Vault with inactivity parameter.
3. Registration of the inheritance (heir address and amount).
4. Possibility of check-in to keep the inheritance blocked.
5. Claim by the heir after the defined period.
6. Early cancellation by the creator if desired.

5. Security and Cryptographic Assumptions

- The smart contract is audited to ensure its immutability.
- There is no custody by **JXLabs** or third parties.
- Inactivity verification is based on the absence of "Check-in" transactions.
- Standard security mechanisms from **Metamask** and the RSK network are used.

6. Limitations and Risks

- The loss of the heir's access to their wallet invalidates the claim.
- Errors when entering the heir address cannot be corrected once the Vault is registered.
- The inheritance cannot be divided among multiple beneficiaries (for now).

7. Roadmap

- **Q2 2025:**
 - Beta cycle and community testing
- **Q3 2025:**
 - Release **Mainnet** on **Rootstock**
 - Interface optimization and user experience improvements.
- **Q4 2025:**
 - Strategic alliances with projects in the **Rootstock** ecosystem.
 - Participation in Bitcoin events and hackathons.

8. Legal Disclaimer

BALT is a decentralized system that does not offer financial or custodial services. Users are fully responsible for their private keys and the addresses used when interacting with the platform. **JXLabs** does not have access to the funds nor can it intervene at any time on the deployed contracts.

9. Equipment

BALT is a development of **JXLabs**, an independent research and development lab on Bitcoin and its second-layer ecosystem.

9.1 Contact:

- Web: <https://jxlabs.xyz>
- Email: ping@jxlabs.xyz
- GitHub: <https://github.com/jxlabsOK>
- X: <https://x.com/jxlabs>