ROBIN/FLOW
FINANCE

GENESIS PARTICIPATION

WHITEPAPER

Follow the flow.
Own the proof.

VERSION 1.0
SEPTEMBER 2026
PUBLIC PRODUCT DOCUMENT

Contents

This document defines the limited Version 1 participation model, user protections, contract behavior, product direction, and risks.

Follow the flow. Own the proof.

Important Notice

Robinflow Finance is an early-stage, self-custodial Web3 participation product. Version 1 focuses on a transparent wallet-based participation record and a simple points system.

Flow Points are non-transferable records. They are not tokens, money, securities, equity, debt, deposits, claims on treasury assets, or a promise of future value. Robinflow Finance does not promise an airdrop, token allocation, financial return, access right, reward, or other benefit.

The optional 0.01 BOT boost is a voluntary protocol participation payment. It is not an investment, deposit, staking position, purchase of a financial product, or payment for a promised future return. Network gas is separate.

This document describes the intended product design. Deployed contract code, verified addresses, and the live interface govern actual behavior. Users should independently review transaction details before signing.

1. Abstract

Web3 products often ask users to trust a page, a database, or a campaign operator before the user can verify what actually happened. Robinflow Finance takes a smaller and more disciplined first step: it lets a wallet create a permanent participation record through a public smart contract.

A user connects a compatible self-custodial wallet and may claim 100 Flow Points through a zero-value contract call. The user pays only network gas for the base claim. After the base claim, the user may choose to submit one exact 0.01 BOT boost transaction and receive an additional 100 Flow Points. A wallet that completes both actions holds 200 Flow Points.

Each wallet may claim once and boost once. The contract enforces these limits. The interface must not award points before a successful transaction receipt, fabricate a transaction hash, or substitute browser storage for contract state.

Version 1 is intentionally narrow. It establishes a verifiable participation layer and a design foundation for future finance-oriented discovery tools. It does not provide trading, lending, staking, yield, portfolio management, custody, brokerage, or investment advice.

2. Why Robinflow Finance Exists

On-chain markets are fragmented across interfaces, contracts, data providers, and communities. Users frequently encounter three problems:

1. **Participation is difficult to verify.** A campaign may show a badge or balance that exists only in a private database.
2. **Product claims outrun product reality.** Early websites often present advanced finance capabilities, growth figures, or reward expectations before those systems exist.
3. **Wallet interactions are poorly explained.** Users may not understand whether a transaction transfers value, grants approval, creates custody risk, or pays only gas.

Robinflow Finance addresses these problems by beginning with a transparent and limited contract model. The first release makes the product's boundaries visible:

- the base action is a one-time zero-value contract call;
- the optional boost is one exact 0.01 BOT payment;
- points are stored as non-transferable records;
- no token approval is requested;
- no asset deposit is required;
- no trading permission is granted;
- no future reward is promised.

This foundation allows later product layers to be added only when their data, permissions, risks, and failure states can be shown accurately.

3. Product Principles

3.1 Proof before promotion

A successful state should be backed by a confirmed transaction receipt and readable contract state. Robinflow Finance must not use random hashes, delayed animations, or local counters to imitate on-chain success.

3.2 Self-custody by default

Users keep control of their wallets and assets. Version 1 does not request token approvals, move user-held tokens, open positions, or hold deposited funds.

3.3 Narrow scope before feature expansion

The initial product solves one complete problem instead of pretending to be a full financial platform. Future modules remain conceptual until they can be built with reliable sources and clear user protections.

3.4 Clear optionality

The base claim and paid boost are distinct actions. The boost must never be preselected, hidden in a combined fee, or described as necessary to keep base points.

3.5 Honest uncertainty

No public copy may imply that Flow Points will become a token, convert into money, guarantee eligibility, or create a contractual claim. Future possibilities must remain conditional.

3.6 Minimal data collection

The public contract requires only a wallet address and transaction data. The website should avoid collecting personal data that is not necessary for operation, security, analytics, or legal compliance.

4. Version 1 User Flow

The primary user journey contains four stages.

Stage 1 - Connect

The user connects a compatible EVM wallet. Connecting exposes the wallet address and permits the application to request signatures or transactions, but it does not itself transfer assets or award points.

The interface reads the current contract state for that address:

- whether the base claim has been completed;
- whether the optional boost has been completed;
- the current Flow Points balance.

Stage 2 - Claim 100

An eligible wallet may call the base claim function. The function accepts no payment. The wallet pays only the network gas required to include the transaction.

After confirmation, the contract records:

- the wallet address;
- a completed base claim;
- 100 Flow Points;
- updated aggregate participation counters;
- a claim event and timestamp.

The same wallet cannot claim again.

Stage 3 - Review the optional boost

After the base claim, the interface may present the optional boost. Before the wallet opens, the review state must show:

- the exact payment: 0.01 BOT;
- estimated network gas when available;
- points before the action: 100;
- points after confirmation: 200;
- the receiving treasury address in a inspectable detail view;
- a clear statement that the action is optional;
- a clear statement that no future benefit is promised.

Stage 4 - Boost to 200

The user may submit one boost transaction with exactly 0.01 BOT. The contract forwards the payment to the configured treasury in the same transaction. If the transfer fails, the entire action reverts and no boost points are added.

After confirmation, the contract records:

- the completed boost state;
- an additional 100 Flow Points;
- a final wallet balance of 200 Flow Points;
- updated aggregate boost and payment counters;
- a boost event and timestamp.

The same wallet cannot boost again.

5. Flow Points

Flow Points are a simple contract-based participation record.

5.1 Fixed rules

Version 1 uses fixed values:

- base claim: 100 Flow Points;
- optional boost: 100 additional Flow Points;
- maximum Version 1 balance per wallet: 200 Flow Points;
- base claims per wallet: one;
- boosts per wallet: one;
- boost payment: exactly 0.01 BOT.

5.2 What Flow Points are

Flow Points indicate that a wallet completed one or both Version 1 participation actions. They can be displayed in the interface and read directly from the contract.

5.3 What Flow Points are not

Flow Points are not:

- an ERC-20 asset;
- transferable between wallets;
- withdrawable;
- redeemable for cash;
- a deposit balance;
- ownership in Robinflow Finance;
- governance power;
- interest or yield;
- proof of a unique human identity;
- guaranteed eligibility for any future program;
- a promise of a token or airdrop.

5.4 Address-level participation

The contract applies rules to wallet addresses. One wallet is not necessarily one person, and one person may control multiple wallets. Robinflow Finance does not present the participation count as a verified count of unique humans.

5.5 Future utility

Robinflow Finance may explore additional uses for participation records, but no future use is committed by this document. Any later program would require separate rules, disclosures, technical review, and user communication.

6. Optional 0.01 BOT Boost

The boost exists to provide a second, explicit on-chain action for users who choose deeper participation.

6.1 Voluntary action

A user can keep the 100 base points without boosting. The product must not create urgency, punish non-payment, or imply that payment guarantees future value.

6.2 Exact amount

The contract accepts exactly 0.01 BOT for the boost. A lower or higher amount reverts. Users cannot enter an arbitrary amount.

6.3 Gas disclosure

Network gas is separate from the 0.01 BOT payment. The interface should show both values when an estimate is available and should never describe the boost as a total all-inclusive cost.

6.4 Treasury handling

The boost payment is forwarded to an immutable treasury address in the same transaction. The contract is not designed to accumulate user boost payments. The production treasury must be approved by the project owner before deployment and should use an appropriate operational security model, such as a project-controlled multisignature wallet.

6.5 Failed transfer behavior

If the treasury cannot receive the payment, the complete transaction reverts. The user remains unboosted, no additional points are issued, and aggregate counters remain unchanged.

7. Smart Contract Model

The Version 1 registry is designed as a small, auditable state machine.

7.1 Public state

The contract exposes:

- base claim status by wallet;
- boost status by wallet;
- Flow Points by wallet;
- total participating wallets;
- total completed boosts;
- total Flow Points issued;
- total BOT processed through boosts;
- pause state;
- owner and pending owner;
- immutable treasury address.

7.2 User functions

The intended user functions are:

- `claimGenesis()` - claim the fixed base allocation with no value;
- `claimAndBoost()` - claim and boost atomically with exactly 0.01 BOT;
- `boostGenesis()` - boost after an earlier base claim with exactly 0.01 BOT;
- `pointsOf(address)` - read a wallet's point balance;
- `accountState(address)` - read claim, boost, and point state together.

7.3 Administrative functions

The owner may pause or unpause participation and may initiate a two-step ownership transfer. The owner cannot:

- change an individual wallet's points;
- create a second claim or boost for a wallet;
- change the fixed Version 1 point amounts;
- change the fixed boost price;
- replace the immutable treasury;
- transfer user-held assets;
- mint a token through this contract.

7.4 Direct transfers

Direct BOT transfers and calls with unsupported data revert. Users must use the intended claim or boost functions so that payment and point state remain synchronized.

7.5 Events

The contract emits events for base claims and boosts. Events include the wallet, points awarded, payment where applicable, and timestamp. Interface history must come from real indexed events or another verified data source, not fabricated sample records.

8. Transaction Integrity and Interface Rules

A trustworthy contract can still be undermined by an inaccurate interface. Robinflow Finance therefore defines a strict transaction state model.

8.1 Required states

The interface should distinguish:

- disconnected;
- connected but unsupported network;
- eligible to claim;
- awaiting wallet confirmation;
- transaction submitted;
- pending confirmation;

- confirmed;
- rejected by user;
- reverted;
- replaced or cancelled;
- RPC unavailable;
- contract unavailable;
- already claimed;
- eligible to boost;
- already boosted.

8.2 Receipt-based success

The interface may display success only after a successful receipt is returned and contract state is refreshed. A submitted hash is not final confirmation.

8.3 Pending recovery

A transaction hash should be saved locally only as a recovery hint. When the user returns, the application should query its status and reconcile it with contract state. Local storage must never become the source of truth for points.

8.4 Real links only

Explorer links must use the actual transaction hash and configured explorer base. If a hash or explorer is unavailable, the interface must show an honest unavailable state rather than a placeholder link.

8.5 No hidden approvals

Version 1 should never request an ERC-20 approval, permit signature, allowance increase, token deposit, or trading authorization.

9. Security Model

9.1 Contract safeguards

The reference contract uses:

- one-time claim and boost checks;
- exact payment validation;
- checks-effects-interactions ordering;
- reentrancy protection around payable actions;
- immediate payment forwarding;
- pause controls;
- two-step ownership transfer;
- immutable treasury configuration;
- custom errors;

- reverting receive and fallback functions.

9.2 Production controls

Before production release, the project must verify:

- compiler and dependency versions;
- full automated test coverage for critical paths;
- deployed bytecode and source match;
- owner and treasury addresses;
- front-end contract address and ABI;
- supported network configuration;
- explorer links;
- successful base and boost transactions;
- pause and ownership procedures;
- incident and rollback documentation.

9.3 Wallet safety

Users should review the destination, value, and requested method before signing. Robinflow Finance cannot protect users from compromised devices, malicious wallet extensions, phishing sites, leaked keys, or transactions signed outside the official interface.

9.4 No absolute guarantee

Smart contracts, wallets, RPC providers, browsers, and hosting systems may fail or contain vulnerabilities. Security controls reduce risk but do not eliminate it.

10. Data, Privacy, and Analytics

10.1 Public data

Wallet addresses, transaction hashes, contract events, timestamps, and point balances are public blockchain data. They may be observed and analyzed by third parties.

10.2 Website data

The website may process limited technical information needed for operation, fraud prevention, performance, and analytics, such as:

- browser and device information;
- approximate region inferred from network data;
- page views and interaction events;
- error and performance logs;
- connected public wallet address;
- transaction status requests.

The production privacy notice must name the actual analytics, hosting, wallet, and infrastructure providers in use. It must not list providers that are not deployed.

10.3 No unnecessary account requirement

The Version 1 core flow should not require a centralized username or password unless the final template and product owner introduce a justified account feature. Wallet state remains the primary participation identity.

10.4 Data accuracy

Aggregate metrics should be read from contract counters or a verified indexer. When data cannot be retrieved, the interface should show an unavailable or stale state rather than silently converting failure into zero.

11. Product and Design Architecture

Robinflow Finance combines a compact pixel-influenced identity with a strict template-first implementation process.

11.1 Brand idea

The logo turns an R built from a pixel grid into a forward route. It represents a wallet entering a visible path, creating proof, and moving toward future finance discovery.

11.2 Public experience

The primary page should communicate the entire product without exaggeration:

1. clear brand and participation status;
2. a large statement of purpose;
3. base claim and optional boost actions;
4. a four-step explanation;
5. verifiable account state;
6. future direction described as research, not a live service;
7. whitepaper, privacy, terms, and risk disclosures.

11.3 Template fidelity

The final website must be built inside the exact template supplied by the project owner. Header, hero proportions, grids, containers, typography hierarchy, spacing, components, motion, and responsive behavior should remain recognizably derived from that template. The implementation should pass a weighted fidelity score of at least 90 out of 100 before production release.

11.4 Independent identity

The reference site's compact headline, status strip, hard-edged controls, and four-step rhythm may inform the design method. Robinflow Finance must use original naming, logo, copy, icons, media, and code, and must not imply affiliation with the reference project or its rights holders.

12. Future Direction

Robinflow Finance may later explore tools that help users understand on-chain financial activity. Possible research areas include:

- wallet-level activity summaries;
- protocol and asset discovery;
- transaction context and risk labels;
- watchlists and saved research;
- evidence-linked market summaries;
- readable contract and position explanations;
- agent-ready structured context.

These are directions, not Version 1 services. Robinflow Finance will not present a feature as live until its data sources, permissions, latency, coverage, errors, and user risks are implemented and disclosed.

Future finance features should preserve several boundaries:

- read-only analysis before execution;
- no custody by default;
- no automatic trading without separate explicit authorization;
- source and timestamp attached to material data;
- risk and invalidation shown with conclusions;
- no fabricated market, user, or performance figures.

13. Roadmap

Phase 1 - Genesis participation

- original Robinflow Finance brand;
- responsive public website;
- wallet connection;
- base claim and optional boost;
- points and account state;
- transaction recovery;
- public whitepaper and legal pages;
- verified production deployment.

Phase 2 - Participation history

Subject to infrastructure validation:

- verified account activity history;
- aggregate participation views;

- improved status and notification controls;
- accessibility and internationalization improvements.

Phase 3 - Read-only finance discovery

Subject to reliable data providers and product review:

- asset and protocol discovery;
- source-linked summaries;
- wallet context views;
- watchlists and research sessions.

Phase 4 - Extended ecosystem tools

Only after separate technical, security, legal, and user-experience review:

- additional evidence layers;
- integrations for agents and applications;
- advanced risk and monitoring features.

Roadmap items may change, be delayed, or never launch. They do not create obligations or entitlements for Flow Point holders.

14. Governance and Operations

Version 1 does not use token voting or a DAO. Operational control rests with the designated contract owner and project treasury.

The project should document:

- who controls the owner account;
- who controls the treasury;
- whether multisignature protection is used;
- pause criteria;
- incident communication channels;
- ownership transfer procedure;
- contract and front-end release history.

A production deployment should not use an individual developer's personal wallet as an undocumented owner or treasury. Changes to the website must not misrepresent unchanged contract behavior.

15. Risks

Users should understand the following non-exhaustive risks.

Smart contract risk

Code may contain defects, dependency issues, or unforeseen behavior. A successful audit or test suite cannot guarantee the absence of vulnerabilities.

Network and infrastructure risk

Transactions may be delayed, dropped, replaced, repriced, or unavailable. RPC, explorer, indexer, wallet, hosting, and domain services may fail.

Treasury risk

Boost payments are sent to the configured treasury. Users should treat the payment as final unless applicable law requires otherwise. Flow Points do not represent a claim on treasury funds.

Wallet risk

Users are responsible for private-key security and transaction review. Compromised devices, malicious software, phishing, or user error may cause loss.

Regulatory risk

Rules affecting digital assets, payments, websites, privacy, and promotions differ by jurisdiction and may change. Access or features may be restricted.

Product risk

Robinflow Finance is early-stage. Features may change, pause, migrate, or discontinue. Future directions may not launch.

Interpretation risk

A points balance records contract participation only. It should not be interpreted as unique identity, reputation, investment value, or guaranteed eligibility.

16. User Responsibilities

By using Robinflow Finance, users should:

- access the correct official domain;
- use a wallet they control;
- verify transaction destination and value;
- understand that the base claim still requires gas;
- understand that the 0.01 BOT boost is optional and separate from gas;
- avoid signing unexpected approvals or permissions;
- retain transaction records when needed;
- comply with applicable laws and wallet-provider terms;
- stop and seek independent advice when they do not understand an action.

17. Conclusion

Robinflow Finance begins with a simple proposition: participation should be visible, limited, and verifiable.

A wallet may claim 100 Flow Points and may voluntarily add one 0.01 BOT boost for another 100. The smart contract enforces the rules. The interface waits for real confirmation. The user keeps custody. The points carry no promised financial value or future entitlement.

This disciplined foundation is more important than a long list of unbuilt features. Robinflow Finance can expand only when each new capability is supported by real infrastructure, clear permissions, accurate data, and honest disclosure.

Follow the flow. Own the proof.

Disclaimer

This whitepaper is provided for general product information and does not constitute investment, financial, legal, tax, or other professional advice. Robinflow Finance does not promise a token, airdrop, profit, return, redemption, or future benefit. Digital-asset and smart-contract interactions involve risk. Users are responsible for their wallets, transaction decisions, legal compliance, and independent evaluation.