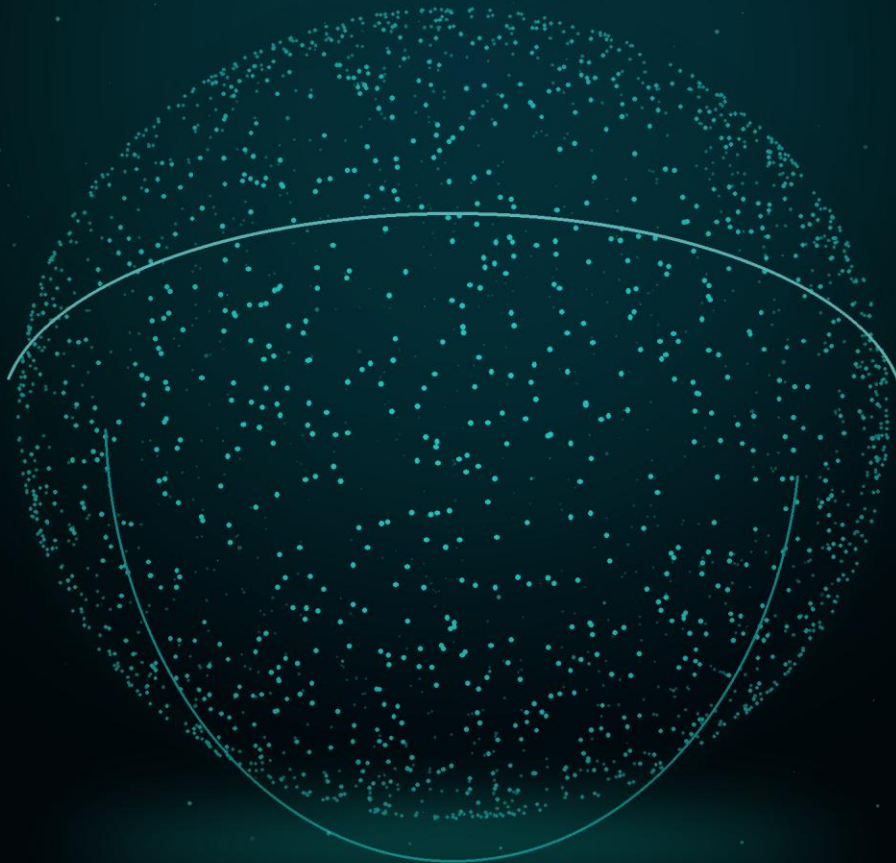




TECHNICAL WHITEPAPER

High-Performance Decentralized
Trading Infrastructure



VERSION 1.0 | JULY 2026

AVARAX.IO

CONTENTS

A structured view of the Avarax protocol,
trading infrastructure and future ecosystem.

01 Executive Summary

02 Vision & Mission

03 The Market Challenge

04 The Avarax Solution

05 Platform Overview

06 System Architecture

07 Trading Engine & Order Book

08 Markets, Orders & Position Modes

09 Margin, Risk & Liquidations

10 Funding, Insurance Fund & ADL

11 Liquidity & Market Data

12 On-Chain Settlement & Custody

13 Security & Operational Resilience

14 Fees & Economic Model

15 API & Automation Ecosystem

16 Copy Trading, Bots & Portfolio Tracking

17 Roadmap & Future Development

18 Governance, Legal & Risk Disclosures

01 / EXECUTIVE SUMMARY

Built for professional trading. Designed for user control.

Avarax is a next-generation decentralized perpetual trading protocol designed to combine high-performance execution with transparent, user-controlled infrastructure. Its objective is to deliver the precision and functionality expected from professional trading venues without requiring users to surrender control to a traditional centralized intermediary.

01 THE CHALLENGE

Centralized exchanges can provide speed and liquidity, but typically require custodial trust. Many decentralized venues improve transparency while still facing limitations in execution quality, advanced order handling, capital efficiency or user experience.

02 THE AVARAX RESPONSE

Avarax separates time-sensitive trading operations from final settlement. Orders are signed by users and processed through a high-performance off-chain matching layer, while deposits, withdrawals and settlement are anchored to smart-contract infrastructure.

03 THE OUTCOME

The result is a modular trading environment built around three principles: execution efficiency, verifiable market operations and user control. This architecture is intended to support professional tools without abandoning the transparency and openness of decentralized finance.

PROTOCOL AT A GLANCE

A modular foundation for the next generation of markets.

Avarax is structured as an integrated protocol rather than a single trading interface. The execution engine, risk controls, settlement layer, market data services and user-facing applications are designed as distinct but coordinated components that can evolve without compromising the integrity of the system.

01 Hybrid Execution

An off-chain order book and matching engine are designed for responsive execution, batch fills and partial fills, while smart contracts provide the settlement and asset-control foundation.

02 Professional Trading

The platform is designed to support market, limit, stop, IOC, FOK, post-only, OCO, trailing, reduce-only and close-only logic, together with one-way and hedge position modes.

03 Integrated Risk

Pre-trade validation, cross and isolated margin, funding, liquidation, an insurance fund and auto-deleveraging are intended to protect market integrity during normal and stressed conditions.

04 Verifiable Settlement

Users connect through external self-custodial wallets. Signed instructions, smart-contract vaults and on-chain settlement records are designed to reduce dependence on opaque custodial processes.

05 Automation Ecosystem

REST and WebSocket interfaces are planned to support algorithmic trading, copy trading, trading bots, analytics services, portfolio tracking and future mobile applications.

06 Sustainable Economics

The protocol is intended to generate revenue primarily through transparent trading fees and a modest withdrawal fee, aligning growth with genuine platform usage rather than mandatory token dependence.

PUBLICATION PRINCIPLE

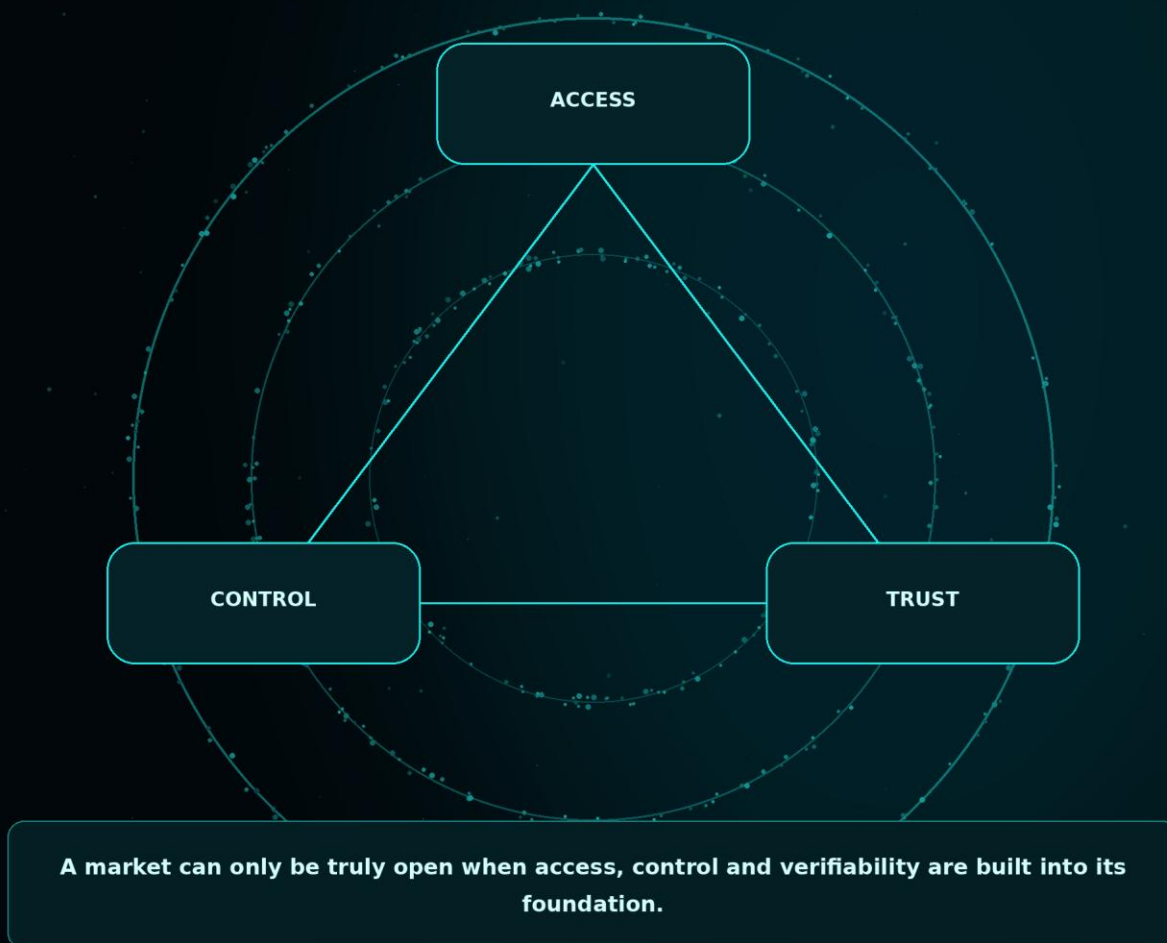
Architecture, supported markets, fees and performance targets remain subject to testing, security review and launch approval. Public metrics will be stated only after they have been validated under representative operating conditions.

02 / VISION & MISSION

Trade Beyond Limits.

Our vision is to create a global trading environment in which sophisticated market access is no longer defined by geography, closed infrastructure or the need to surrender control to an intermediary.

Avarax envisions an open financial layer where professional execution, verifiable settlement and intuitive technology operate as one coherent system. Traders should be able to access advanced tools, manage risk and participate in global markets through infrastructure that is transparent by design and built for continuous evolution.



OUR MISSION

Build infrastructure that earns trust through design.

Our mission is to engineer a high-performance decentralized trading environment that unites professional execution, robust risk management and verifiable settlement in a platform that remains accessible, intuitive and user-controlled.

01 Performance with Purpose

Optimize execution and reliability around real trading needs, not headline metrics alone.

02 Transparency by Design

Make platform mechanics, fees, risk controls and settlement logic understandable and verifiable.

03 Control without Friction

Preserve self-custodial access while creating an experience that remains efficient and intuitive.

04 Professional Tools for All

Bring advanced order handling, margin controls and analytics to a broader global user base.

05 An Open Ecosystem

Enable APIs, automation, copy trading, bots and portfolio services through controlled, documented access.

06 Responsible Evolution

Scale carefully through testing, security review, monitoring and evidence-based product decisions.

THE AVARAX COMMITMENT

Avarax is not intended to be a decentralized copy of a centralized exchange. It is being built as market infrastructure that uses decentralization where trust matters and performance where execution matters.

03 / THE MARKET CHALLENGE

Trading infrastructure still forces a compromise.

Digital-asset traders are commonly asked to choose between two incomplete models: centralized platforms optimized for execution and decentralized platforms optimized for verifiability. Each model solves important problems, yet neither consistently delivers the complete combination required by professional and automation-driven market participants.

CENTRALIZED VENUES

Often optimized for throughput, liquidity access and familiar trading workflows.

+ Responsive execution

+ Mature trading interfaces

+ Established liquidity access

- Custodial dependence

- Opaque internal controls

- Counterparty exposure

DECENTRALIZED VENUES

Often optimized for open access, transparent settlement and user-controlled wallets.

+ Self-custodial access

+ Verifiable settlement

+ Open participation

- Execution constraints

- Fragmented liquidity and data

- Limited professional tooling

THE UNRESOLVED GAP

The market still needs an architecture that combines professional execution with transparent rules, verifiable settlement and meaningful user control.

STRUCTURAL BARRIERS

Six challenges shaping the next generation of trading.

The opportunity is not created by one isolated weakness. It emerges from a series of connected infrastructure gaps that affect execution, trust, capital efficiency and the ability to build reliable trading workflows.

01 Custodial Dependence

When a venue controls private keys and internal account balances, users must trust its solvency, security practices, withdrawal policies and operational governance.

02 Execution Limitations

Many decentralized venues can struggle to reproduce responsive order-book behavior, advanced order logic and predictable execution during fast markets.

03 Fragmented Liquidity & Data

Liquidity, trades, candles and order-book data may be distributed across networks and venues, complicating price discovery and consistent analytics.

04 Incomplete Risk Transparency

Funding, margin, liquidation, insurance and deleveraging rules are often difficult for users to evaluate before those mechanisms become critical.

05 Restricted Automation

Inconsistent APIs, permissions and data streams can limit safe access for algorithms, copy trading, bots, portfolio systems and external analytics.

06 Experience & Access Friction

Wallet workflows, network complexity and limited professional interfaces can create barriers for traders moving from established platforms into Web3 markets.

MARKET NEED

The market need is clear: traders should not have to choose between performance and sovereignty, or between advanced functionality and transparent infrastructure.

04 / THE AVARAX SOLUTION

One architecture. No false compromise.

Avarax is designed as a high-performance perpetual trading protocol that separates time-sensitive execution from trust-sensitive settlement. The model combines an off-chain order book and matching layer with user-authorized orders, dedicated risk controls and verifiable on-chain custody and settlement.

01 / ACCESS

Self-Custodial Entry

Traders connect an external wallet and retain control of authorization. Signed actions and API permissions establish a secure path into the protocol.

02 / EXECUTION

Order Book & Matching

A dedicated execution layer manages price-time priority, advanced order logic, partial fills and responsive market workflows.

03 / CONTROL

Pre-Trade Risk Engine

Margin, leverage, position limits and reduce-only or close-only rules are evaluated before an order can affect protocol state.

04 / SETTLEMENT

On-Chain Assurance

Smart-contract custody and settlement are intended to provide verifiable balances, transparent transfers and a scalable path toward rollup-based settlement.

Avarax applies decentralization where trust matters and specialized execution infrastructure where speed, order logic and market responsiveness matter.

FROM CHALLENGE TO SYSTEM DESIGN

Six barriers. One coherent response.

The Avarax architecture is designed as an integrated system: execution, risk, custody, data and user experience reinforce one another rather than operating as disconnected product layers.

01 Custody

External-wallet access and smart-contract custody are designed to reduce reliance on opaque internal account control.

02 Execution

An off-chain order book and matching engine support advanced orders, partial fills and professional trading workflows.

03 Liquidity & Data

Unified order-book, trade, candle and WebSocket interfaces create a consistent foundation for market access and analytics.

04 Risk

Pre-trade controls, cross and isolated margin, funding, liquidation, insurance and ADL form a layered risk framework.

05 Automation

API keys plus planned REST and WebSocket interfaces enable bots, copy trading, portfolio tools, analytics and algorithmic strategies.

06 Experience

A professional interface, grouped positions, transparent history and familiar wallet flows reduce friction across desktop and future mobile access.

DESIGN PRINCIPLE

Avarax is not designed to put every interaction on-chain. It is designed to make authorization, custody and settlement verifiable while preserving the responsiveness required by modern derivatives markets.

05 / PLATFORM OVERVIEW

A complete trading stack, built as one system.

Avarax brings the core functions of perpetual trading into a unified platform. The interface, execution engine, risk framework, account system, market data and settlement layer are designed to operate as coordinated parts of the same protocol.

01 Trading Experience

A professional workspace combines charts, order entry, order-book depth, positions, PnL, funding and account history.

DESKTOP

FUTURE MOBILE

02 Execution & Orders

The order-book engine supports Market, Limit, IOC, FOK, Post-Only, Stop, OCO, Trailing, Reduce-Only and Close-Only workflows.

PARTIAL FILLS

BATCH MATCHING

03 Positions & Risk

Independent positions can use dedicated TP/SL controls, while cross or isolated margin, leverage, funding and liquidation rules govern exposure.

ONE-WAY / HEDGE

INSURANCE / ADL

04 Data & Connectivity

Markets, tickers, depth, recent trades and multi-timeframe candles form the data layer for the interface and planned external integrations.

REST API

WEBSOCKET

05 Custody & Settlement

External-wallet access, collateral accounting and smart-contract settlement provide the foundation for transparent asset movement and verifiable state.

SELF-CUSTODIAL

ON-CHAIN

Platform objective: offer the tools expected by experienced traders without disconnecting execution quality from transparent, user-authorized infrastructure.

THE TRADING LIFECYCLE

From wallet connection to verifiable settlement.

The platform journey is designed to remain familiar to active traders while preserving explicit authorization, transparent risk rules and a traceable account history.



PLATFORM SCOPE

Markets	Perpetual trading with order-book execution
Collateral	Initial focus: ETH, USDT, USDC and BNB
Account	Balances, positions, PnL, fees, rebates and funding
Connectivity	Web interface plus planned API and automation ecosystem

Production performance, liquidity depth and final settlement capacity will be reported only after infrastructure testing, deployment and independent validation.

06 / CORE FEATURES

Control across the entire trade lifecycle.

Avarax is designed around the workflows active traders use every day: precise entry, flexible execution, independent position management and transparent account outcomes—all within one perpetual trading environment.

01 Advanced Order Suite

A broad order model gives traders control over price, timing and execution behavior.

- Market and Limit orders
- IOC, FOK and Post-Only
- Stop Market, Stop Limit, OCO and Trailing

02 Independent Positions

Multiple positions can remain individually manageable instead of being reduced to one opaque net record.

- Dedicated entry, leverage and margin
- Individual Take Profit and Stop Loss
- Manual edit and close controls

03 Execution Engine

The order-book and matching layer is structured to process executable liquidity while preserving order state and fill history.

- Price-level aggregation
- Partial and batch fills
- Order cancel and edit lifecycle

04 Professional Account View

The interface connects active orders, positions and outcomes in a consistent operational view.

- Open, cancelled, filled and closed states
- Realized and unrealized PnL
- Fees, rebates, funding and trade history

TRADER-FIRST DESIGN

Complex infrastructure should produce a clearer trading experience—not a more complicated one. Avarax is built to keep control visible at every decision point.

INFRASTRUCTURE FEATURES

Performance is only useful when the system remains resilient.

Execution is supported by a second layer of core capabilities focused on risk discipline, custody, reliable data access and controlled automation.

05 Layered Risk Framework

Pre-trade checks, leverage limits, cross and isolated margin, funding, liquidation processes, insurance reserves and ADL are designed to work together.

RISK BEFORE EXECUTION

06 Self-Custodial Access

Users connect supported external wallets. Collateral movements and account authorization are designed around explicit user action and verifiable settlement.

USER-AUTHORIZED

07 Market Data Interfaces

Market, ticker, order-book depth, recent trade and candle interfaces provide a consistent data surface for the platform and its integrations.

REST + WEBSOCKET

08 Automation Ecosystem

Controlled API access is planned to support trading bots, copy trading, portfolio tracking, analytics, algorithmic workflows and future mobile products.

PLANNED ECOSYSTEM

OPERATIONAL SAFEGUARDS

- **Signed authorization** Orders and protected actions are linked to verified user or API permissions.
- **Rate and abuse controls** Limits, wallet controls and freeze mechanisms protect sensitive flows.
- **Monitoring and recovery** Health, security-event and recovery functions support operational visibility.
- **Auditable history** Orders, fills, funding, fees and balance changes remain traceable.

The feature set describes the intended Avarax product architecture. Production throughput, latency, liquidity and security claims require deployment testing and independent validation before publication.

07 / TRADING ENGINE ARCHITECTURE

From authorized order to consistent state.

The Avarax trading engine separates order intake, risk validation, matching, accounting and market-data publication into clearly defined stages. This limits cross-system side effects and makes each state transition easier to test and audit.

01 Order Gateway

Normalizes wallet, symbol, side, order type, size, price, leverage, time-in-force and position-control flags.

02 Authorization

Validates signed user instructions or controlled API permissions before an order enters execution.

03 Pre-Trade Risk

Checks balance, margin, leverage, position limits and reduce-only or close-only constraints.

04 Symbol Queue

Routes the instruction to an isolated per-market book and serializes conflicting matching actions.

05 Matching Core

Applies crossing rules and price-time priority, then creates full or partial fills against available liquidity.

06 State & Ledger

Updates orders, positions, balances, margin, PnL, fees and funding records through traceable state changes.

07 Data Publication

Publishes book snapshots, trades, tickers and candles without allowing the read layer to mutate execution state.

08 Settlement Path

Transfers finalized outcomes toward the protocol ledger and intended verifiable on-chain settlement layer.

STATE TRANSITION PRINCIPLE

An accepted order does not imply a completed trade. Each instruction progresses through explicit states—open, partially filled, filled, cancelled or rejected—with fills and account effects recorded independently.

This separation supports reliable order history, deterministic reconciliation and controlled recovery when execution is interrupted.

ENGINE DESIGN PRINCIPLES

Deterministic by design. Scalable by separation.

The engine architecture prioritizes predictable rules and isolated responsibilities. Scaling is pursued through market separation, controlled concurrency and optimized fill processing rather than through hidden changes to execution behavior.

01 Price-Time Priority

Bids rank by highest price and earliest arrival; asks rank by lowest price and earliest arrival.

FAIR QUEUE ORDER

02 Symbol Isolation

Each market maintains an independent book and execution lock, reducing cross-market race conditions.

PER-MARKET CONTROL

03 Fill Efficiency

Price-level aggregation, partial fills and batch processing reduce unnecessary work while preserving fill detail.

OPTIMIZED PROCESSING

04 Time-in-Force Logic

GTC, IOC and FOK instructions define whether unfilled quantity remains open, cancels immediately or requires full execution.

EXPLICIT OUTCOMES

05 Read/Write Separation

Market-data endpoints read books and persisted trades without modifying matching, custody, risk or authorization state.

SAFE DATA LAYER

06 Invariant Monitoring

Book health, ledger checks, deterministic stress routes and recovery hooks help detect inconsistent state.

TESTABLE OPERATIONS

VALIDATION STANDARD

The architecture has deterministic test and monitoring paths, but production throughput and latency targets will only be published after deployment-scale load testing and independent validation.

08 / ORDER TYPES & EXECUTION

Choose how the trade enters—and how it exits.

Avarax combines immediate execution, price-controlled orders and conditional position management. Each order type has a defined trigger, matching behavior and remaining-quantity outcome.

01 Market

Executes immediately against available book liquidity. The final price may span multiple levels.

IMMEDIATE EXECUTION

02 Limit

Executes at the specified price or better. Unmatched quantity follows the selected time-in-force.

PRICE CONTROL

03 Stop Market

Activates after the stop trigger is reached, then submits a market instruction.

PROTECTIVE TRIGGER

04 Stop Limit

Activates after the stop trigger and submits a limit order at the configured price.

TRIGGER + PRICE LIMIT

05 Take Profit Market

Closes eligible exposure with a market instruction once the take-profit level is crossed.

EXIT PRIORITY

06 Take Profit Limit

Activates a limit exit after the take-profit trigger, preserving price control.

CONTROLLED EXIT

07 OCO

Links two conditional orders so execution or cancellation of one removes the paired instruction.

ONE CANCELS THE OTHER

08 Trailing Stop

Adjusts its trigger as the market moves favorably, then activates when price reverses by the defined distance.

DYNAMIC PROTECTION

Conditional orders remain subject to trigger availability, market liquidity, risk checks and the execution rules active when they are released.

EXECUTION CONTROLS

Explicit rules. Predictable outcomes.

Execution controls determine whether an order rests, cancels, adds liquidity or may only reduce an existing position. These instructions are validated before matching.

GTC

Good-Til-Cancelled

Unfilled quantity remains open until filled or cancelled.

IOC

Immediate-or-Cancel

Fills available quantity immediately and cancels the remainder.

FOK

Fill-or-Kill

Requires the full requested quantity or cancels without a fill.

POST

Post-Only

A resting limit instruction that must not execute immediately as taker.

REDUCE

Reduce-Only

May decrease eligible exposure but must not increase or reverse it.

CLOSE

Close-Only

Restricts the instruction to closing an existing position.

POSITION-AWARE EXECUTION

- **Independent positions** Each Avarax position can maintain its own entry, leverage, TP and SL.
- **TP/SL replacement** Saving new protection replaces the prior active TP/SL for that position.
- **Add-trade behavior** Adding exposure does not automatically recreate or duplicate TP/SL orders.
- **Manual control** Eligible open orders and positions can be edited, cancelled or closed explicitly.

ORDER LIFECYCLE

OPEN

PARTIALLY FILLED

FILLED

CANCELLED

REJECTED

Execution records also retain outcome reasons such as TP_HIT, SL_HIT, MANUAL_CLOSE and TRADE for transparent account history.

Order execution remains dependent on available liquidity, risk validation and the state of the relevant market at the time of matching.

09 / RISK MANAGEMENT & LIQUIDATIONS

Risk is managed before it becomes a liquidation.

Avarax applies layered controls across order entry, margin, pricing, account health and operational security. The objective is to contain exposure early and preserve orderly market operation under changing conditions.

01 Pre-Trade Validation

Orders are checked against available balance, margin requirements, leverage limits, position constraints and reduce- or close-only rules before matching.

02 Cross Margin

Eligible collateral can support multiple positions within the same margin account, allowing gains and available equity to offset losses across exposure.

03 Isolated Margin

Collateral is assigned to a specific position, limiting the amount directly exposed to that position while keeping its health calculation separate.

04 Price Integrity

Mark, index and oracle inputs—including freshness and confidence signals—are designed to reduce reliance on a single volatile or unavailable price source.

05 Funding, Fees & PnL

Funding transfers, trading fees and realized and unrealized PnL feed into the account state used for ongoing margin and health evaluation.

06 Wallet & Abuse Controls

Risk scoring, rate limits, monitoring and wallet freeze or unfreeze controls help contain suspicious behavior and protect critical protocol operations.

ACCOUNT HEALTH

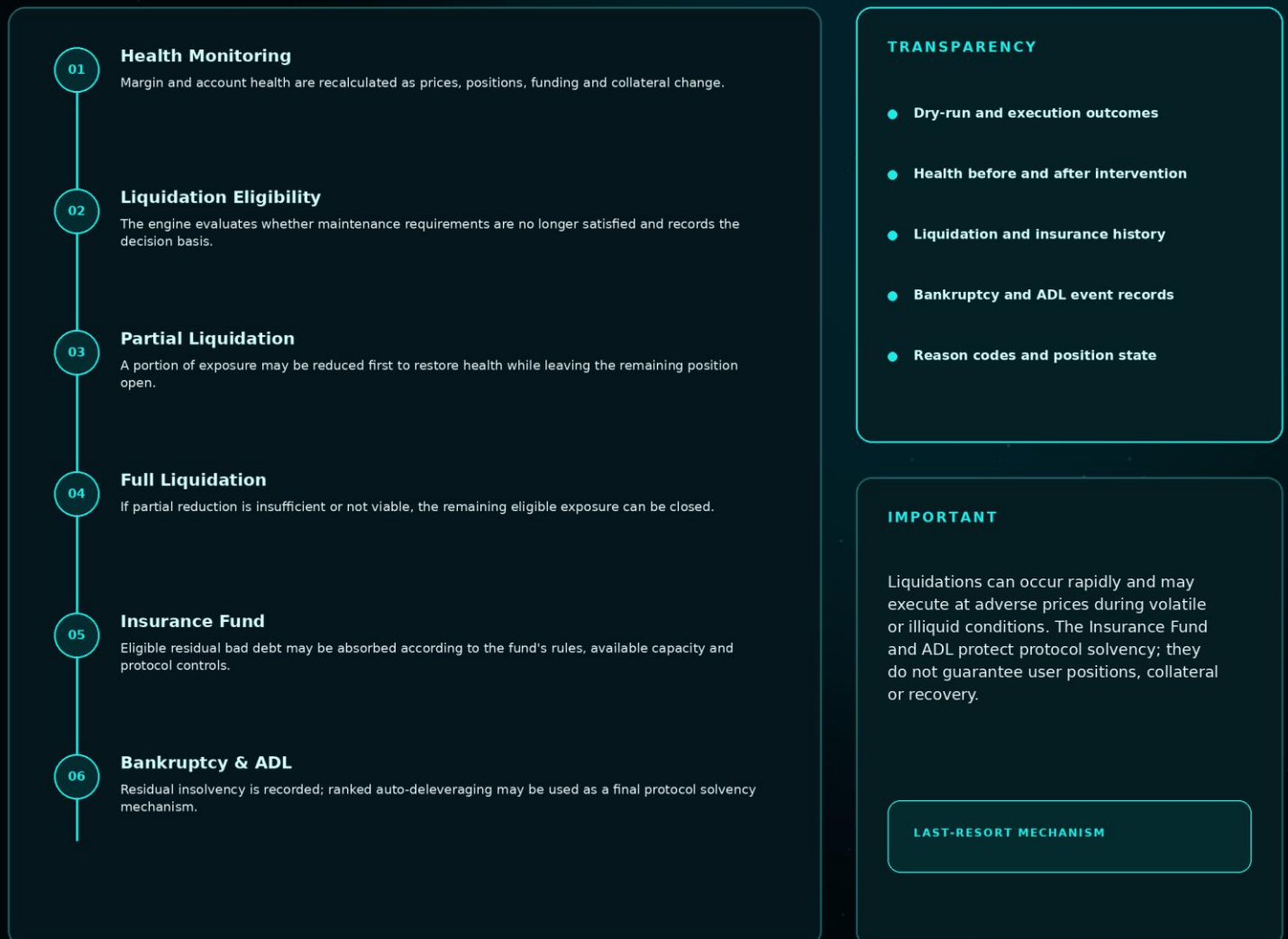
Account health reflects the relationship between eligible collateral and the maintenance requirements of open exposure. When health deteriorates, protective actions can progress from order restrictions to position reduction and liquidation.

Risk controls reduce protocol and market exposure; they cannot eliminate volatility, execution risk or trading losses.

LIQUIDATION WATERFALL

Controlled intervention. Transparent escalation.

Liquidation is an ordered risk process—not a single event. Avarax is designed to restore account health with the least disruptive eligible action before escalating to protocol-level safeguards.



Partial liquidation is designed to reduce exposure and restore health before full closure is considered—supporting a more measured response to margin stress.

10 / LIQUIDITY & MATCHING

Liquidity becomes useful when execution is reliable.

Avarax is designed around a transparent central limit order book. Liquidity is organized by price level, exposed through market-data interfaces and consumed according to deterministic matching rules.

CLOB Order Book

Bids and asks are maintained per market. Buyers compete from the highest price, sellers from the lowest, with remaining quantity visible by level.

DEPTH Price Levels

Orders at the same price are grouped into depth levels, creating an aggregated view of available size without changing individual priority.

MAKER Market Makers

Professional and algorithmic participants can quote both sides of a market, subject to risk, balance and order-validation requirements.

TAKER Market Takers

Incoming marketable orders consume the best available opposite-side liquidity and may execute across multiple price levels.

PARTIAL Partial Fills

If the full requested quantity is unavailable at eligible prices, execution can occur in portions while order state and remaining size are updated.

DATA Market Data

Read-only feeds provide markets, ticker, depth, trades, candles and WebSocket updates for interfaces, analytics and trading systems.

LIQUIDITY PRINCIPLES

- **Visible depth** Aggregated bids and asks show the available size currently resting in the book.
- **No invented fills** Orders execute only against eligible liquidity or an explicitly defined execution source.
- **State consistency** Orders, fills, positions and market-data events are updated from the same execution outcome.
- **Open participation** The architecture is intended to support multiple liquidity sources without granting hidden priority.

IMPORTANT DISTINCTION

Avarax provides the infrastructure for organized liquidity; it cannot guarantee continuous depth, a specific spread or execution at the displayed top-of-book. Available liquidity changes with participant activity and market conditions.

MATCHING PROCESS

Deterministic priority. Consistent state.

The matching engine processes eligible orders within a symbol-isolated queue. Price determines competitiveness; time determines priority among orders resting at the same price.

01 Validate

Confirm signature, order parameters, balance, margin, leverage and market state before execution.

02 Route

Direct the instruction by market and order type. Market orders do not rest in the order book.

03 Select Liquidity

A buy consumes the best asks; a sell consumes the best bids. Limit orders cross only at their price or better.

04 Apply Priority

Within each eligible price level, older resting orders are selected before newer orders.

05 Create Fills

Generate one or more fills, calculate weighted average execution and update remaining quantity and status.

06 Commit & Publish

Apply trade, fee, position and margin changes, then publish trade and order-book updates to clients.

PERFORMANCE ARCHITECTURE

- **Symbol isolation** Separate matching queues reduce cross-market contention.
- **Batch fills** Related fills and state changes can be accumulated and committed together.
- **Price aggregation** Depth is maintained efficiently by price level while retaining order priority.
- **Race protection** Serialized market queues prevent competing updates from corrupting book state.

EXECUTION OUTCOMES

FILLED

PARTIALLY FILLED

RESTING

CANCELLED / REJECTED

Market orders consume available liquidity immediately and never enter the book. Unfilled market quantity is cancelled or rejected when no eligible liquidity remains.

11 / USER EXPERIENCE

Professional tools. Clear decisions.

Avarax is designed to make complex perpetual trading understandable without removing professional control. Market context, order entry, risk information and position actions remain visible within one coherent workspace.

01

Market Context

Chart, market selector, mark price, funding, depth and recent activity provide the context required before an order is submitted.

02

Focused Order Entry

Side, order type, price, size, leverage, margin mode and execution controls are grouped into a single validated workflow.

03

Independent Positions

Each position retains its own entry, size, leverage and protective settings, supporting deliberate management of separate trades.

04

Position-Level TP/SL

Take-profit and stop-loss controls can be added or replaced directly from the relevant position without duplicating protection.

05

Actionable Status

Open, partially filled, filled, cancelled, rejected and closed states communicate what happened and what the user can do next.

06

Transparent History

Separate order and trade histories show execution, side, size, price, result, leverage, time and outcome reasons.

DESIGN PRINCIPLES

- **Clarity before density** Critical information is prioritized; secondary detail remains available on demand.
- **Confirm material actions** Orders, cancellations, position edits and closes show the intended consequence.
- **Consistent terminology** The same labels, states and reasons are used across order, position and history views.
- **Responsive by design** Layouts are intended to preserve core controls across desktop and mobile form factors.

NON-CUSTODIAL CONTROL

The interface helps users understand and execute actions; it does not take control of their decisions. Wallet authorization, visible risk parameters and explicit confirmation preserve user agency throughout the trading experience.

TRADING WORKFLOW

From intent to informed oversight.

The user journey is structured around progressive disclosure: connect, assess, configure, confirm, monitor and review. Each stage reveals the information needed for the next decision without separating the user from market context.

STEP 01

Connect & Verify

Connect a supported wallet, verify network and permissions, and review available collateral before entering the trading workspace.

STEP 02

Assess the Market

Select a market and review chart, mark price, order-book depth, funding and recent trades before defining the instruction.

STEP 03

Configure the Order

Choose side, order type, quantity, leverage, Cross or Isolated margin and any time-in-force, reduce-only or close-only control.

STEP 04

Review & Submit

Preview material order details and risk parameters, then authorize the instruction. Validation feedback explains any rejected action.

STEP 05

Manage the Position

Monitor PnL and health, edit eligible orders, add or replace position-level TP/SL, cancel instructions or close exposure manually.

STEP 06

Review the Record

Use Order History and Trade History to inspect status, fills, fees, leverage, timestamps and reasons such as TP_HIT, SL_HIT or MANUAL_CLOSE.

POSITION-AWARE CONTROLS

EDIT

Modify eligible open-order parameters.

TP / SL

Add or replace protection for the selected position.

CANCEL

Remove an eligible open instruction.

CLOSE

Explicitly reduce or close the selected position.

USER FEEDBACK

BEFORE

Show parameters, estimated consequence and applicable risk checks.

DURING

Display pending, open, partially filled or processing state without ambiguity.

AFTER

Confirm the final status, remaining quantity, position impact and recorded reason.

12 / API & DEVELOPER ECOSYSTEM

Built for interfaces. Designed for automation.

Avarax exposes protocol capabilities through clearly separated API surfaces. Public market data supports discovery and analysis; authenticated interfaces enable account-specific actions without exposing user secrets.

PUBLIC Public Market Data Read-only endpoints provide markets, ticker, order-book depth, recent trades, candles and protocol health information.	REAL-TIME WebSocket Streams Market channels distribute incremental book, trade and price updates for responsive interfaces and automated monitoring.	SIGNED Trading API Authenticated routes support order submission, OCO and trailing instructions, cancellation and access to eligible account data.
PRIVATE Account & History Authorized clients can retrieve orders, positions, fills and history that belong to the wallet bound to the API credential.	PROTECTED Security Controls Timestamp, nonce and HMAC signature validation reduce replay and tampering risk; rate and risk controls apply before execution.	RESTRICTED Operational APIs Recovery, security and administrative functions remain separated from public developer access and require elevated authorization.

SIGNED REQUEST MODEL

x-api-key	Identifies the credential and its wallet scope.	x-timestamp	Limits the accepted time window for a request.
x-nonce	Provides uniqueness and helps prevent replay.	x-signature	Authenticates method, path, time, nonce and request body.

CREDENTIAL SAFETY

API documentation and endpoint schemas may be public. API secrets, wallet keys and administrative credentials must never be published, embedded in client-side code or shared through downloadable documentation.

DEVELOPER ECOSYSTEM

Integrate securely. Build with confidence.

The Avarax developer ecosystem is intended to support trading terminals, analytics, portfolio tools, market makers and institutional integrations through versioned documentation and controlled access.

API Reference

AVAILABLE

Curated documentation describes public market data, authenticated trading routes, request schemas, signing and response behavior.

WebSocket Documentation

AVAILABLE

Channel definitions explain market subscriptions, event payloads, reconnect behavior and client-side state synchronization.

API-Key Management

AVAILABLE

Users can create, list and disable credentials. Permissions and operational limits can evolve with production access policies.

SDKs & Examples

PLANNED

Official libraries and reference clients can simplify signing, validation, subscriptions and common order-management workflows.

Sandbox / Testnet

PLANNED

A controlled environment can allow developers to test integrations without relying on production balances or live market activity.

GitHub & Bug Bounty

PLANNED

Selected tools, examples and security scopes may be published alongside a responsible disclosure and reward program.

INTEGRATION LIFECYCLE

01 DISCOVER 02 TEST 03 AUTHORIZE 04 INTEGRATE 05 MONITOR 06 ROTATE / REVOKE

Developers should begin with read-only data, validate in a non-production environment, then enable only the minimum permissions required by the integration.

RESPONSIBLE ACCESS

- **Least privilege** Grant only the scopes an integration requires.
- **Secret isolation** Store credentials server-side or in a secure secret manager.
- **Operational resilience** Handle rate limits, reconnects, duplicate events and degraded data.
- **Independent verification** Confirm order and position state after every material action.

13 / FUTURE FEATURES

Extend the platform. Preserve the core.

Avarax is designed as a modular trading ecosystem. Future capabilities can expand market access, automation and portfolio intelligence while retaining the same principles of non-custodial control, transparent execution and measured risk.

01 Copy Trading

Users may be able to follow eligible strategy providers with configurable allocation, leverage caps, stop conditions and transparent performance data.

02 Trading Bots

Rule-based and API-connected automation may support recurring, conditional and market-making strategies within defined permissions and risk limits.

03 Portfolio Tracker

A unified view may aggregate collateral, open exposure, realized and unrealized PnL, fees, funding and historical performance.

04 Spot & Swap

A separate spot experience may support direct asset exchange without coupling the interface or failure domain to perpetual trading.

05 Advanced Analytics

Trader dashboards may add attribution, drawdown, risk-adjusted performance, execution-quality and strategy-comparison insights.

06 Mobile Trading

A dedicated mobile experience may prioritize monitoring, protective actions, alerts and secure authorization without replicating desktop density.

AUTOMATION WITH USER CONTROL

- **Explicit opt-in** No strategy or bot should trade without deliberate authorization.
- **Defined limits** Users should be able to cap allocation, leverage, loss and permitted markets.
- **Visible attribution** Every automated action should identify its source, outcome and position impact.
- **Immediate revocation** Users should be able to pause a strategy and revoke its credentials.

FORWARD-LOOKING STATEMENT

These capabilities describe potential product directions, not guaranteed releases. Scope, timing and availability may change following technical validation, security review, market conditions and applicable legal requirements.

EXPANSION FRAMEWORK

Progress through validated releases.

Future modules should move from research to controlled testing and only then to general availability. The sequence prioritizes infrastructure, safety and observable behavior before feature breadth.

Automation Foundation NEAR TERM SDKs, reference clients, granular API permissions, alerts and sandbox tooling can establish the base for reliable automation.	Portfolio Intelligence NEAR TERM Portfolio tracking, execution analytics, risk summaries and exportable reporting can improve decision support and oversight.
Copy-Trading Pilot MID TERM A controlled cohort can test provider verification, performance methodology, allocation controls and follower-protection mechanisms.	Bot Marketplace MID TERM Approved strategy integrations may be discoverable through transparent permissions, risk labels, version history and revocable access.
Spot / Swap Module LONG TERM A separate trading surface can extend market access while preserving operational separation from the perpetual engine.	Institutional Connectivity LONG TERM Advanced reporting, subaccounts, delegated roles and specialized connectivity may support professional trading operations.

RELEASE GATES

01 TECHNICAL 02 SECURITY 03 RISK 04 LEGAL 05 PILOT 06 OBSERVABILITY

A feature advances only when its behavior is testable, its risks are understood and the platform can monitor, limit and reverse operational impact.

DESIGN COMMITMENTS

- Modular deployment**

New products should not destabilize the core perpetual engine.
- Backward compatibility**

Versioned interfaces should protect existing integrations where practical.
- Measured rollout**

Feature flags, limits and staged access should contain early operational risk.
- Transparent status**

Documentation should identify experimental, pilot and generally available features.

14 / ROADMAP

Build the foundation. Validate before scale.

The Avarax roadmap is milestone-driven rather than date-driven. Each phase must demonstrate technical integrity, controlled risk and observable behavior before the protocol advances to broader access.

PHASE 01

Core Infrastructure

FOUNDATION BUILT

Develop the hybrid architecture: off-chain order book and matching, balances and ledger, multi-asset vault testing, positions, PnL, fees, funding and market-data services.

PHASE 02

Trading & Risk Completion

IN DEVELOPMENT

Finalize advanced order paths, Cross and Isolated margin, price integrity, liquidation, Insurance Fund, ADL, recovery, synchronization and comprehensive failure handling.

PHASE 03

Testnet & Validation

PLANNED

Connect production-grade market data and charting, deploy a controlled public testnet, run load, race-condition, recovery and economic simulations, and resolve observed defects.

PHASE 04

Security & Launch Readiness

PLANNED

Complete independent smart-contract and infrastructure reviews, remediation, operational runbooks, monitoring, status communications, legal readiness and liquidity onboarding.

PHASE 05

Public Perpetual Launch

PLANNED

Release the non-custodial perpetual trading experience through app.avarax.io with staged market access, conservative limits and measurable reliability criteria.

RELEASE STRATEGY

Expand deliberately. Measure continuously.

Following a stable perpetual launch, Avarax can expand through controlled product and market phases. Later milestones remain conditional on the performance, security and regulatory requirements observed in preceding stages.

PHASE 06 API & Automation

Introduce production API access, granular permissions, SDKs, reference clients, sandbox tooling and monitored bot integrations.

PHASE 07 Portfolio Intelligence

Add portfolio tracking, risk summaries, execution analytics, alerts and exportable performance reporting.

PHASE 08 Copy-Trading Pilot

Test strategy-provider verification, performance methodology, follower controls, allocation limits and transparent risk labels.

PHASE 09 Market Expansion

Evaluate additional perpetual markets and a separate Spot/Swap module without compromising core-engine isolation.

PHASE 10 Institutional Scale

Explore subaccounts, delegated roles, advanced reporting, specialized connectivity and professional liquidity programs.

CONTINUOUS Protocol Hardening

Maintain audits, monitoring, incident response, recovery testing, documentation, bug bounty and performance optimization.

MILESTONE GATES

01 CORRECTNESS

02 SECURITY

03 RESILIENCE

04 LIQUIDITY

05 COMPLIANCE

06 OBSERVABILITY

Advancement requires evidence that the system can execute, recover and communicate reliably under the expected operating conditions.

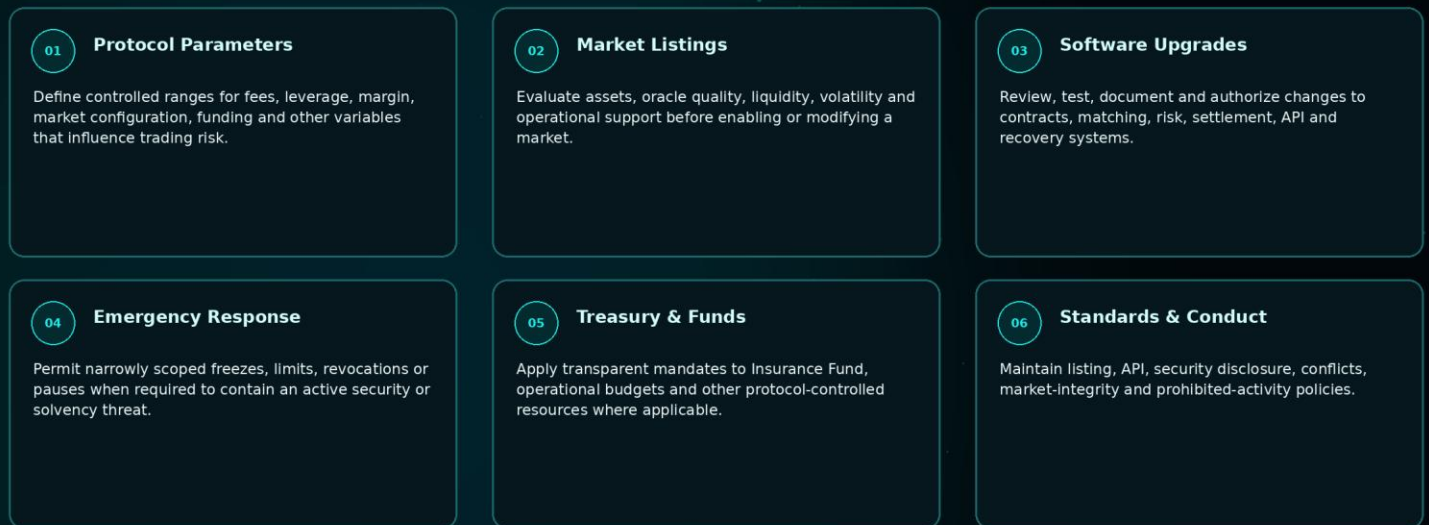
ROADMAP DISCLAIMER

This roadmap communicates current development intentions and is not a promise of delivery, timing or availability. Priorities may change in response to testing, security findings, market conditions, resources and applicable law.

15 / GOVERNANCE

Accountability before decentralization.

Avarax governance is intended to protect protocol integrity while the system matures. During early stages, clearly accountable operators can respond to incidents, validate changes and maintain legal and operational responsibility.



CHANGE CONTROL

01 PROPOSE 02 ASSESS 03 TEST 04 APPROVE 05 DEPLOY 06 MONITOR

Material changes should retain an identifiable proposer, review record, test evidence, approval authority, deployment outcome and post-release monitoring trail.

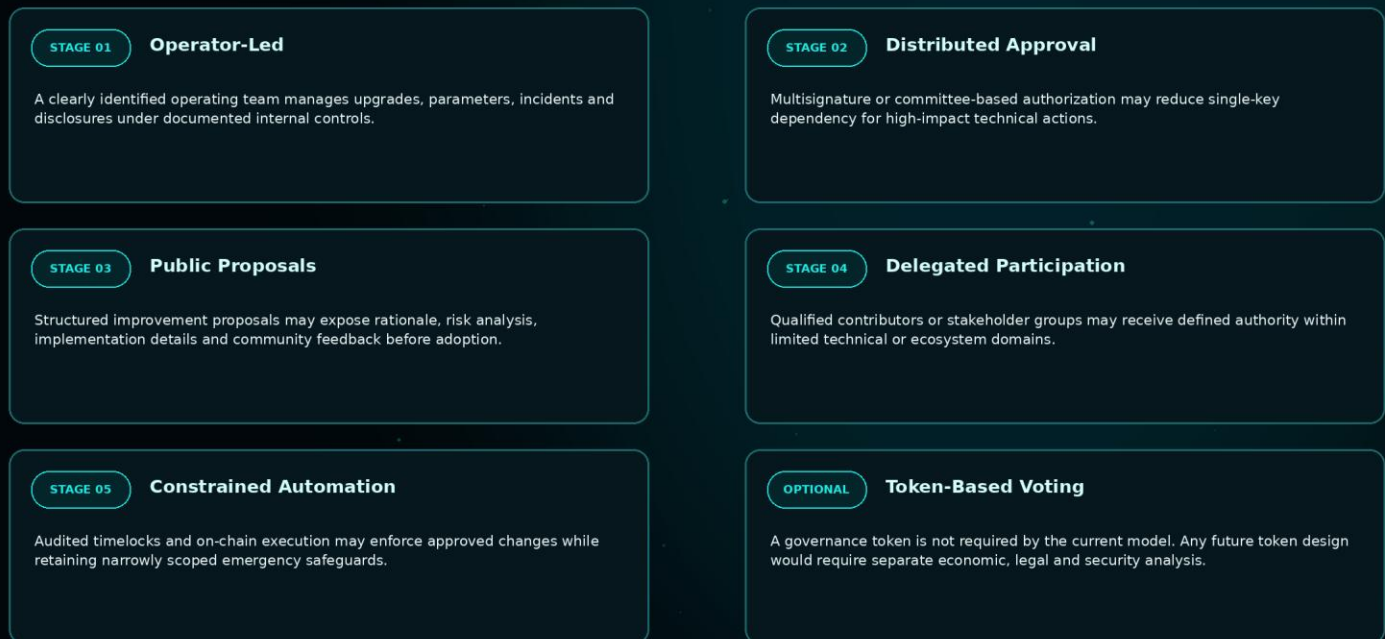
GOVERNANCE BOUNDARY

Protocol governance does not override smart-contract constraints, user ownership, contractual obligations or applicable law. Governance participation does not guarantee profit, compensation, voting power or influence over every operational decision.

PROGRESSIVE GOVERNANCE

Decentralize with evidence—not assumptions.

Governance may evolve as the protocol, contributor base and security model mature. Authority should be distributed only when responsibilities, incentives, technical constraints and emergency procedures are sufficiently defined.



SAFEGUARDS

- Least authority** Each role receives only the powers required for its mandate.
- Time to review** Material non-emergency changes should allow adequate notice and scrutiny.
- Conflict disclosure** Decision-makers should disclose material interests where relevant.
- Revocable roles** Delegated authority should be measurable, limited and removable.

NO IMPLIED RIGHTS

This section describes a possible governance evolution. It does not create voting, ownership, dividend, redemption or other legal or economic rights, and it does not commit Avarax to launch a DAO or governance token.

16 / OPTIONAL TOKEN UTILITY

Utility before speculation.

Avarax does not require a native token for its current protocol design. If a token is considered in the future, it should solve clearly defined coordination or utility problems and remain subordinate to security, legal clarity and product usefulness.

01 Governance Access

A token could support participation in defined protocol proposals without creating ownership, dividend or unrestricted operational rights.

02 Fee-Tier Utility

Eligible activity or holdings could qualify users for transparent fee tiers, subject to limits, availability and applicable law.

03 Service Bonding

Market makers, strategy providers or infrastructure operators could post bonded value against narrowly defined service obligations or misconduct rules.

04 Ecosystem Incentives

Time-limited programs could reward measurable contributions such as liquidity, development, education or security research.

05 Developer Programs

Utility may support grants, integrations or contributor initiatives governed by published eligibility and accountability standards.

06 Community Coordination

Token-based signaling could help prioritize non-binding ecosystem proposals while accountable operators retain required legal duties.

UTILITY PRINCIPLES

- **Optional** Core trading, custody and settlement should not depend on speculative demand.
- **Functional** Each utility should have a measurable purpose beyond marketing or price support.
- **Transparent** Eligibility, limitations, emissions and conflicts would require clear disclosure.
- **Revocable** Programs should be adjustable or terminable when risk, abuse or law requires it.

NO TOKEN ANNOUNCEMENT

This section is exploratory. It is not an offer, sale, solicitation, airdrop, listing announcement or commitment to create a token. No person should rely on it as an expectation of value, profit, liquidity or future availability.

TOKEN EVALUATION FRAMEWORK

Necessity first. Design only if justified.

A future token decision would require a separate design and disclosure process. The burden is to demonstrate that a token improves protocol coordination without introducing disproportionate legal, economic, security or user risks.

01 Demonstrated Need

Identify the exact function that cannot be achieved more safely or simply through existing assets, permissions or contractual arrangements.

02 Legal Analysis

Assess classification, offering restrictions, disclosures, jurisdictional access, sanctions, taxation and consumer-protection requirements.

03 Economic Sustainability

Model incentives, concentration, circulating supply, dilution, treasury dependence and adverse behavior under stressed conditions.

04 Security Review

Audit contracts, minting and administrative controls, bridges, governance execution and any staking or bonding mechanism.

05 Distribution Integrity

Define transparent eligibility, conflicts, vesting, transfer restrictions and protections against manipulation or unfair allocation.

06 Operational Readiness

Establish monitoring, incident response, disclosures, support and procedures for upgrades, suspension or termination of utility programs.

CURRENTLY NOT DEFINED

NAME / SYMBOL

NOT DEFINED

SUPPLY

NOT DEFINED

ALLOCATION

NOT DEFINED

SALE

NOT PLANNED

AIRDROP

NOT PLANNED

LISTING

NOT ANNOUNCED

No token contract, tokenomics, issuance schedule or distribution commitment is established by this whitepaper.

NO ECONOMIC RIGHTS

Any potential token would not automatically represent equity, ownership, debt, revenue participation, dividends, redemption rights or a guarantee of platform access, exchange listing, liquidity, price stability or appreciation.

17 / LEGAL DISCLAIMER

Information only. Risk remains yours.

This whitepaper describes the current Avarax concept and intended development direction. It is provided for general information and must not be treated as a guarantee, binding commitment or substitute for independent professional advice.

01 No Offer or Solicitation

Nothing in this document constitutes an offer, invitation or solicitation to buy, sell or subscribe for any asset, token, security or service.

02 No Financial Advice

Avarax does not provide investment, trading, legal, tax or accounting advice. Users must make and verify their own decisions.

03 No Guarantee

No representation is made that the protocol, roadmap, features, markets, liquidity, performance, availability or outcomes will meet expectations.

04 Forward-Looking Statements

Plans and technical descriptions may change without notice due to testing, security findings, resources, markets or applicable law.

05 Eligibility & Restrictions

Access may be unavailable in certain jurisdictions or to sanctioned, restricted or otherwise ineligible persons.

06 Terms Control

Use of any released service is governed by the applicable Terms of Service, risk disclosures and policies, which prevail over this whitepaper.

NO WARRANTIES

To the maximum extent permitted by applicable law, this whitepaper and any described technology are provided on an “as is” and “as available” basis without express, implied or statutory warranties, including accuracy, completeness, merchantability, fitness for a particular purpose, non-infringement, uninterrupted operation or error-free performance.

OWN-RISK ACKNOWLEDGEMENT

Any use of Avarax, connected wallets, APIs, smart contracts or third-party services is voluntary and at the user's sole risk. Users remain responsible for eligibility, wallet security, credentials, transactions, positions, taxes and compliance obligations.

MATERIAL RISK DISCLOSURES

Loss may be immediate and total.

Digital-asset and leveraged perpetual trading involve substantial risk. Users should participate only if they understand the mechanics, can bear the full loss of committed assets and can independently evaluate changing market conditions.

01 Market & Leverage

Volatility, gaps, funding, fees and leverage can amplify losses and trigger liquidation with limited or no time to react.

02 Liquidity & Execution

Orders may fill partially, at adverse prices or not at all. Displayed depth, spreads and market access can change rapidly.

03 Smart Contracts

Bugs, exploits, upgrade failures, oracle faults or network conditions may cause loss, delay, incorrect state or unavailable functionality.

04 Wallet & Key Security

Lost keys, compromised devices, malicious signatures, phishing or incorrect addresses may result in irreversible asset loss.

05 Infrastructure

APIs, nodes, databases, relayers, interfaces, internet access or third-party services may fail, become delayed or provide inconsistent data.

06 Regulatory Change

Laws, classifications, sanctions, licensing, taxation or enforcement may restrict access, features, markets or continued operation.

07 Liquidation Safeguards

Partial liquidation, Insurance Fund and ADL are protocol solvency tools; they do not guarantee user collateral or reimbursement.

08 Third Parties

Wallets, bridges, oracles, networks and integrations are independently operated and may introduce risks beyond Avarax's control.

LIMITATION PRINCIPLE

To the fullest extent permitted by law, Avarax and relevant contributors, affiliates and service providers disclaim liability for losses arising from reliance on this whitepaper or use of the platform. Rights that cannot lawfully be excluded remain unaffected.

Before publication or launch, this disclaimer and the complete legal framework should be reviewed by qualified counsel for every jurisdiction in which access is offered.