Tradient Whitepaper

AI trading infrastructure for Robinhood Chain

Version 1.0 | 6 November 2026

Tradient connects compatible AI assistants to market data, portfolio research and controlled trade execution. Users bring their preferred AI, retain ownership of their wallet and define what that AI is permitted to do.

The platform combines a remote Model Context Protocol server, a web control dashboard, an execution policy engine and narrowly scoped signing credentials. The same controls apply whether an instruction originates in an AI conversation or an authorized background automation.

This whitepaper defines the operating model, transaction lifecycle, product modules, economic rules and release standards. It is a product and architecture specification. A capability becomes available only after its integration, security and market-access requirements pass the release gates described here. These specifications do not represent certification that an integration or audit has already been completed.

The commercial focus is an English-first experience for self-directed traders and developers in eligible European markets. Availability is determined separately for crypto, tokenized equities, lending, bridging, derivatives and exchange connections.

Name: Tradient

Token identifier: $TRADIENT

Settlement network: Robinhood Chain

Primary interface: the user's connected AI assistant

Control interface: the Tradient web dashboard

Reading guide

The whitepaper separates the user's operating experience from the technical conditions that make that experience possible. The appendices define the tool families, records and release evidence required for implementation.

Sections	Subject
1 and 2	Product scope, eligible audience and the end-to-end user journey
3 and 4	Service architecture, wallet ownership and signing modes
5 to 7	Permissions, consent, quotes and reliable execution
8 and 9	Market data, research and portfolio accounting
10 to 14	Automations, lending, bridging, perps and exchange accounts
15 and 16	MCP interoperability, fees and the token boundary
17 and 18	Security, operating standards and release gates
Appendices A to D	Tool catalog, data records, acceptance scenarios and references

Operating principles

The AI interprets user intent and selects permitted tools. Tradient validates inputs, calculates amounts, enforces limits and constructs transactions. Numerical accounting and authorization never depend on a language model's judgment.

User consent has a defined object: a particular quote, a particular automation mandate or a bounded delegation policy. A statement generated by the AI is not evidence that the user approved an action.

Wallet ownership, delegated signing and external exchange custody are different arrangements. The dashboard identifies which arrangement applies to each balance and action. The platform does not describe an exchange balance as self-custodied.

The design reuses existing wallet and protocol infrastructure. It does not require Tradient to operate its own exchange, lending pool or bridge. Smart accounts and third-party protocols still introduce contract dependencies. A route that cannot be safely constrained remains available only through owner confirmation, or remains disabled.

1 Product scope and audience

Tradient gives a general-purpose AI assistant a controlled operating interface for trading. A user can ask for a portfolio review, inspect a token, obtain a trade quote and authorize execution within one conversation. Tradient provides the tools and execution infrastructure; the connected client supplies the conversational model.

The platform serves three customer groups. Self-directed traders use natural language to manage a bounded allocation. Advanced users configure repeatable execution rules such as DCA and portfolio rebalancing. Developers integrate the same tool interfaces into their own agents, with every connection bound to an identified user and an explicit permission set.

Western market positioning

The initial commercial audience is English-speaking users in eligible European jurisdictions, with EEA markets as the first market-access candidates. Messaging, documentation and support use international English. Portfolio reporting is USD-first; timezone and display-currency preferences do not alter settlement assets or trading permissions.

Country selection is a release decision, not a claim that every European resident qualifies. Each country and product combination requires a documented eligibility assessment, provider acceptance and any required authorization. The launch country list is published before onboarding opens and is versioned when it changes.

Robinhood's Stock Token documentation identifies restrictions involving U.S. persons and jurisdictions including the United States, United Kingdom, Canada and Switzerland. Tradient therefore excludes these groups from its Stock Token launch audience. Crypto-only access and other modules have separate eligibility decisions; availability of one module never unlocks another. [2]

Defined product boundary

Tradient provides execution infrastructure and factual account information. Users choose objectives, assets and permissions. The platform does not promise returns, discretionary portfolio management, an AI-generated investment mandate or universally available access to financial products.

The core release includes account setup, portfolio and price reads, spot quotes, confirmed execution, notifications and revocation. Automations, Earn, bridging, exchange connections and perps have separate activation gates. A provider's support for a network is insufficient by itself to enable a feature.

The onboarding target is a first testnet trade within ten minutes for a user who already has test funds and a compatible AI client. Real-world funding, eligibility checks and external approvals may take longer. Release communications distinguish testnet access, limited mainnet access and general availability.

2 User journey and control dashboard

The user completes setup on the dashboard, then connects an AI assistant. The dashboard remains the place to inspect activity, grant authority, approve exceptions and recover control independently of the AI client.

Step	Concrete user action and result
1 Register	Create an account, declare residence, complete required checks and select a timezone. The enabled module list reflects eligibility.
2 Establish ownership	Create a passkey-owned smart account or use an existing wallet as the owner of a separate smart account. Add a recovery method before delegation.
3 Fund	Copy the chain-specific deposit address or approve a supported bridge route using the source wallet. Funds are recognized after the configured confirmation threshold.
4 Set rules	Choose allowed assets, trade limits, approval requirements and enabled modules. Sign the resulting policy version.
5 Connect AI	Authorize the client through the OAuth consent flow. The connection receives only the scopes the user selected.
6 Preview and act	Ask for a quote, inspect expected output and all fees, then approve through the applicable confirmation mode.
7 Review or revoke	Inspect receipts, pause execution, cancel mandates or revoke signing credentials. Export activity independently of chat history.

A first trade

For the instruction "Buy $40 of an approved token using USDG," the AI resolves the token identity and requests a quote. Tradient interprets the $40 as the maximum principal spend, displays the fee basis and shows minimum received amounts. The dashboard or confirmation card identifies the connected client, wallet, route, policy and expiry.

In owner-confirmed mode the user signs the specific transaction. In bounded delegation mode a signed user policy may authorize eligible trades without a signature for each one. A chat client confirmation remains an additional interaction control, not an independent wallet security boundary.

The receipt separates requested, submitted and confirmed amounts. If settlement is unresolved, the UI shows pending or reconciling rather than reporting success. If a quote expires during approval, the user receives a fresh quote for confirmation.

Dashboard responsibilities

The dashboard contains Wallet, Rules, AI Connections, Activity, Approvals, Automations, Address Book, Notifications and Recovery. Exchange Accounts appears only when that module is enabled. Actions that expand authority require owner authentication and a signed policy change. A pause can always reduce authority without waiting for the AI to respond.

3 System architecture

Tradient separates intent, authorization and execution. The MCP resource server authenticates the client and validates tool inputs. The policy engine evaluates a canonical action. The signer accepts only an approved execution record. The settlement adapter submits through the relevant wallet or venue infrastructure.

Component	Responsibility
MCP resource server	Tool discovery, input validation, authenticated user context and structured responses
Authorization service	OAuth consent, token issuance, client identity and connection revocation
Web dashboard	Ownership proofs, rules, approvals, recovery and emergency controls
Policy engine	Eligibility, budgets, token identity, price checks, recipient rules and permission intersections
Quote service	Route selection, fee normalization, simulations and immutable quote records
Signing service	Credential isolation, action validation and permission-scoped signatures
Settlement adapters	Smart-account UserOperations, owner-signed transactions and enabled venue orders
Worker service	Scheduled actions, price triggers, retries and expiry handling
Data and accounting service	Chain indexing, reconciliation, cost basis and portfolio valuations
Storage and observability	User records, durable jobs, audit events, monitoring and incident evidence

Request path

A read passes through authentication, scope validation and the data service. A write additionally requires a valid preview, a current policy decision, a budget reservation and the appropriate approval artifact. The worker invokes the same action services used by interactive requests; it cannot bypass policy by calling a signer directly.

The implementation baseline is TypeScript with a supported Node LTS release, the official MCP SDK, schema validation and an EVM client library. The dashboard uses Next.js; Postgres stores authoritative records. Redis is a cache and job-dispatch aid, not the authority for balances, quote use or daily budgets.

Robinhood Chain mainnet uses chain ID 4663 and testnet uses 46630, with ETH as gas token. Production uses authenticated RPC infrastructure and an independent failover. Mainnet and testnet have separate users or environment namespaces, credentials, callback origins and deployment registries. [1]

The release registry pins contract addresses, implementation versions, bytecode checks, entry points, allowed selectors, provider environments and enabled adapters. Deployment names alone are never accepted as proof of compatibility.

4 Wallet ownership and signing modes

The default wallet is a separate smart account funded with the amount the user wants Tradient to manage. Ownership belongs to the user's passkey or connected wallet. The backend does not receive the owner seed phrase or unrestricted owner credential.

Robinhood's documentation describes ERC-4337 support. Tradient evaluates an existing smart-account implementation and permission validator against its actual release requirements. Session-key frameworks support scoped delegation, but each required restriction must be tested in the chosen deployment. [3, 4]

Mode	Authority and availability
Owner confirmed	Every money-moving action requires the user's signature. This is the initial execution baseline and fallback when delegation cannot be constrained.
Bounded delegation	A dedicated backend credential executes only the contract calls, token budgets and time window granted by the user. Enabled separately for each supported action type.
Existing wallet as owner	The connected wallet controls a new Tradient smart account. It retains its ordinary owner role without sharing the main wallet's funds with the AI.
Same-address delegation	Optional later integration using compatible wallet infrastructure. It requires a separate compatibility and security review; it is not a core release dependency.

Credential separation

Each wallet uses a distinct execution credential held in a managed signing service or KMS-backed signer. Perps and exchange integrations use separate credentials and revocation paths. OAuth tokens authorize tool requests; they are not wallet keys and do not replace transaction permission checks.

A server compromise can cause loss through actions that a delegated credential permits. Trade caps, expiry and recipient restrictions limit exposure; the term non-custodial does not mean that delegated trading is risk-free. Restricting withdrawals alone cannot prevent damaging trades or malicious execution within weak route permissions.

Recovery and independent exit

Users register a second passkey or a recovery owner. Recovery changes require owner approval, notification and a defined delay. Tradient cannot replace a missing owner credential through email support alone. If every owner credential is lost, access may be unrecoverable.

The wallet implementation and ownership details are exportable. Users can interact with the wallet through a compatible independent interface, revoke delegated access and transfer assets without relying on the Tradient AI connection. A service outage must not convert Tradient into the only route to the wallet.

5 Permission system and default limits

Effective authority is the intersection of user ownership rules, onchain permission bounds, product eligibility, connection scopes, the current policy version and available budget. A permissive result in one layer cannot override a restrictive result in another. Only the owner can broaden a policy.

Setting	Initial product default
Per-trade principal	Maximum $100 equivalent
Aggregate execution budget	$500 per UTC day and $500 over a rolling 24-hour window; the stricter remaining budget applies
Additional owner approval	Required above $50 principal; every transaction requires a signature in owner-confirmed mode
Slippage	1% default; user-defined hard maximum, initially 1%
Transfers, Earn, bridging, perps, exchange writes	Disabled until individually enabled
Session lifetime	Seven days by default; maximum 30 days before owner renewal
Trading frequency	Ten new trade submissions per hour across the user's clients and worker
Active automations	Maximum 20; each requires an expiry and budget
Address-book activation	24 hours after an owner-signed addition or destination change

The $50 approval threshold sits below the $100 per-trade ceiling, so it is reachable. Approval does not bypass that ceiling: a $150 trade remains blocked until the owner changes the ceiling. Policy changes invalidate outstanding quotes and re-evaluate dependent automations.

Hard permissions and service policies

Onchain restrictions cover enforceable target addresses, selectors, recipients, native value, token parameters, cumulative budgets and expiry. The exact supported set is recorded per wallet deployment. A per-call amount bound is not described as a cumulative spending cap. Gas policies are not token spending policies. [4]

USD valuations, sanctions or geographic controls, price freshness and some daily rules run in the backend. Their enforcement depends on Tradient services. If the selected onchain validator cannot enforce the hard bounds required for an action, that action uses owner confirmation. Wildcard targets and unrestricted administrative calls are excluded from delegated permissions.

Reservations are atomic across AI clients, workers and approval requests. A pending execution consumes its reserved budget until the order is reconciled. Transfers and lending deposits consume their configured outflow budget; sells cannot generate unlimited new daily authority. A missing valuation blocks a USD-limited write instead of assigning zero value.

6 Quotes consent and execution lifecycle

Every operation that can move money starts with a preview. A preview binds the user, wallet, connection, environment, asset addresses, amounts, fee recipient, route, destination, minimum output, policy version and expiry. The resulting quote identifier is opaque, single-use and valid for at most 30 seconds, or less when the provider deadline is shorter.

Execute tools accept only that identifier. They cannot override amounts, change recipients or supply arbitrary transaction data. Before signing, the service verifies quote ownership and freshness, reruns policy checks and validates the actual transaction bytes against the quoted intent.

Approval artifacts

There are three valid authorization paths: a signature over the exact owner-confirmed transaction; a dashboard approval bound to a fresh quote; or an already signed delegation or automation mandate covering the exact action. The backend records the approval object, its expiry and the policy version. An AI message claiming that the user agreed does not create an approval object.

An above-threshold approval request can remain visible longer than a quote. Opening it obtains a current quote and displays the new terms. The user approves those terms, not an old price. Expired quotes return QUOTE_EXPIRED; automatic repricing never becomes silent authorization to execute.

State	Meaning and permitted next step
Quoted	Terms are immutable; request owner approval or execute under an existing mandate
Awaiting approval	No signature or submission; refresh terms before final owner approval
Reserved	Quote is consumed and budget reserved atomically; only one execution record exists
Submitted	Transaction or order identity is recorded; further requests return that identity
Reconciling	Submission outcome is uncertain; investigate before any replacement action
Confirmed or failed	Final accounting is applied once, or unused reservations are released
Expired or cancelled	No new execution can begin from this quote

Example

A $40 trade fits a signed $100-per-trade delegation and remaining $500 budget. Tradient reserves $40 plus the separately budgeted fees, validates the route and signs once. A concurrent request using the same quote receives the existing execution status. A second client cannot reuse the first client's quote.

A $75 trade requires owner approval under the default threshold. A $150 trade is blocked under the initial hard trade ceiling. Neither the model nor a paid subscription can override these rules.

7 Routing settlement and accounting integrity

The spot adapter requests an indicative price for discovery and a firm executable quote for a preview. The selected routing baseline is 0x AllowanceHolder on supported routes. Its documentation lists Robinhood Chain support and distinguishes indicative prices from executable quotes. RWA quoting additionally requires the integrating team's RWA opt-in. [5, 6]

Transaction construction

The executor reads actual token decimals, converts decimal strings into integer base units and validates every returned target against the deployment registry. The allowance spender is checked independently from the transaction destination. Approvals cover the exact authorized amount; residual allowances are cleared when the approved route supports atomic cleanup.

Batching combines allowance and swap where supported. Simulation checks balance changes, expected recipient, token identity, minimum output, fee destination and revert behavior. Native value and gas sponsorship are bounded separately. A simulation is a point-in-time test, not a guarantee about future state.

Delegation requires enforceable checks on the complete route. An allowlisted router with unconstrained nested calldata is insufficient. Recipient redirection, unexpected approvals, administrative calls and arbitrary delegatecall are rejected. If the selected validator cannot constrain a route, the service requires an owner signature for that route.

Idempotency and uncertain outcomes

Execution uses a durable action ID and a database uniqueness constraint on the consumed quote. UserOperation nonces or venue client-order identifiers are allocated once and retained across retries. A timeout causes status reconciliation, not a fresh market order.

Rebroadcasting the identical signed transaction is different from constructing a replacement. Any replacement preserves the same intended action, recipient and budget, records the relationship and rechecks consent validity. A transaction that may already have settled is never replaced by a second independent trade.

Receipts and finality

The service records submitted, included, confirmed and reconciled states separately. It indexes actual token transfers and venue fills rather than treating quoted output as realized output. Chain reorganizations reverse provisional records and trigger reindexing; irreversible reporting uses the release's documented confirmation policy.

Fallback routing uses the same validation path. A provider change that alters the previewed route or economics creates a new quote. It cannot be used as a hidden fallback after the user has approved different terms. If no eligible route exists, the tool returns NO_ROUTE without charging a Tradient execution fee.

8 Market data and research

Each data response identifies the asset by chain and contract address, its source, observation time and whether it is cached. Symbols are display labels. Ambiguous searches return candidates rather than selecting a token because its ticker resembles a familiar asset.

Robinhood's read-only Stock Token APIs provide metadata, corporate actions and underlying-equity prices. The documented REST bid and ask require corporate-action multiplier adjustment to compare with token-equivalent values; onchain feeds may already contain that adjustment. Tradient records the source convention and applies the adjustment exactly once. [7]

Valuation and execution checks

For crypto, a recent pool or reference quote supports portfolio valuation. For stock tokens, valuation uses a documented bid, ask or midpoint convention with multiplier metadata. A displayed reference value is not an executable offer. Market session, asset status, halt information, spread and source age accompany the response.

The initial price-deviation guard blocks a spot write when its normalized reference and executable price differ by more than 2%. For buys the comparison uses a consistent buy-side benchmark; for sells it uses a sell-side benchmark. The calculation excludes separately disclosed fees from one side only if it excludes them from both. Stale, missing or incomparable reference data blocks guarded execution.

A 15-second upstream cache cannot support a promise of tick-level stock alerts. The price monitor reports the source timestamp and evaluates only newly observed source data. Polling faster than a source updates does not make the observation newer. Trigger latency is measured from receipt of a usable update.

Research tools

The AI can compare held assets, explain portfolio changes and summarize supplied market facts. Numerical ratios, balances and fee totals are calculated by deterministic services. The model receives source-linked fields and returns interpretations separately from facts. Unsupported causal explanations are labelled as hypotheses, and missing data remains missing.

Token checks cover verified identity, liquidity, age, concentration data where available and transaction simulation. A low-risk classification is an explanation of observed checks, not certification that an asset is safe. New or unverified tokens are disabled for delegated trading by default.

Historical charts preserve pool identity, sampling interval, missing intervals and any pool changes. The service does not silently combine different pools or multiply underlying share history into token history without recording the transformation.

9 Portfolio and performance accounting

The portfolio combines direct wallet holdings with separately identified Earn, perp and linked-exchange positions. It avoids counting both a deposited asset and the claim representing that deposit as independent wealth. All positions carry source and reconciliation timestamps.

Cost basis and realized results

The accounting baseline is weighted-average acquisition cost per asset and chain. A buy adds principal and attributable execution costs to acquisition cost. A sale removes cost at the pre-sale weighted average; realized P&L equals net proceeds minus the cost removed. Costs are recorded in their original asset and translated using the event-time valuation source.

For illustration, buying ten units for $20 creates a $2 average cost before fees. Selling four units for $12 produces $4 realized profit before fees and leaves six units with $12 cost. The calculation does not infer profitability from the asset's current price alone.

Transfers between a user's identified accounts preserve known basis. External incoming assets with unavailable acquisition history are marked unknown basis; they are not assigned zero cost and counted as pure profit. Gifts, bridging, token migrations, rebasing assets and corporate actions have explicit event types and reconciliation rules.

Portfolio change and cash flows

The daily briefing separates valuation change, realized trading results, yield, fees and net external deposits or withdrawals. Depositing $500 is not $500 of investment performance. When cost or historical price evidence is incomplete, the affected metric is shown as unavailable or partial.

Cross-venue fills are normalized by asset identity, fee currency, settlement time and account. Open orders reserve spendable balances but do not become completed trades until fills arrive. Partial fills update positions incrementally. Cancelled unfilled amounts release reservations after venue confirmation.

Evidence and exports

Each entry links to a transaction hash, venue order or position record and retains raw event references. Users can export holdings, normalized transactions, fee breakdowns, cost-basis adjustments and accounting limitations. Exported figures are analytical records rather than certified tax statements.

Indexer disagreements trigger a reconciliation state. The service compares chain balances, recorded transfers and internal fills before overwriting a position. Manual corrections preserve the original entry, the correction reason, who approved it and the resulting recalculation version.

10 Background automations

Automations execute through Tradient's worker after the AI conversation closes. The owner approves a structured mandate defining action type, assets, size, trigger or schedule, budget, expiry and failure behavior. A mandate cannot be created merely because external content tells the AI to create one.

Automation	Trigger and execution behavior
DCA	Spend a fixed authorized amount on a named schedule. Missed runs are skipped by default rather than accumulated into a larger catch-up trade.
Conditional order	Request a fresh quote after a reference price crosses the specified level. Submission still depends on liquidity, limits and route validation.
Stop loss or take profit	Evaluate the specified reference source and sell an explicitly defined quantity. The trigger level is not a guaranteed fill price.
Rebalance	Compare current weights with owner-defined targets and drift tolerance. Preview the required sequence and execute each leg within the mandate budget.

Job identity and scheduling

Every scheduled occurrence has a unique mandate-and-occurrence key. Durable leases prevent two workers from claiming the same occurrence. Triggered orders latch once; repeated observations cannot submit duplicate sells. The engine checks current holdings before quoting an "all" quantity.

Schedules store an IANA timezone and an explicit daylight-saving policy. Daily and weekly jobs follow local wall-clock time. A missing local time moves to the next valid time; a repeated local time executes once. Monthly schedules state how a missing calendar day is handled. Budget resets remain UTC and rolling-24-hour rules regardless of schedule timezone.

Retries and pauses

Transient data or RPC failures retry at most three times with bounded backoff, only while the mandate and quote remain valid. Unknown submission outcomes enter reconciliation. Region blocks, cap breaches, expired credentials and persistent price deviations pause execution immediately rather than retrying blindly.

Each rebalancing leg is separately accounted for. If a later leg fails, the system records the partial rebalance and pauses for review; it does not claim portfolio-wide atomicity. Venue-native conditional orders are used only after their signing and cancellation behavior is validated.

Renewal does not expand the mandate automatically. Pausing an AI connection pauses the mandates explicitly bound to that connection; the dashboard also offers a wallet-wide pause. Already submitted transactions and venue-native orders require their own status or cancellation handling.

11 Earn through approved vaults

Earn connects eligible users to individually approved vault deployments. The first integration candidate is a USDG-denominated Morpho vault. A protocol name is not a vault approval: the actual address, chain, implementation, curator, underlying asset and withdrawal mechanism are reviewed before activation.

ERC-4626 provides familiar asset and share operations, but integrations must follow the specific vault version. Morpho's current documentation identifies version-specific limits and withdrawal behavior. Tradient tests the deployed vault and uses appropriate version-specific limit calculations rather than assuming all max-function responses have identical meanings. [9]

Deposit lifecycle

The user selects an approved vault and previews an asset amount. The preview shows underlying token, expected shares, minimum accepted shares, fees, current position and liquidity information. The recipient is the user's wallet. Approvals are exact-amount and simulation includes the actual asset and share balance changes.

The owner approves the deposit, or an independently enabled Earn mandate covers it. The executor verifies the minimum-share bound through a supported router or account-level mechanism. If the route cannot enforce an economic minimum, delegated deposits remain disabled; a successful generic simulation alone is insufficient.

Withdrawals and valuation

A partial withdrawal specifies assets required; a full exit targets all redeemable shares and discloses expected assets. The adapter checks available liquidity, queues, penalties and receiver restrictions. A withdrawal preview can report a partial available amount without implying that the whole position is liquid.

Positions are valued from redeemable claims using a documented share-to-asset conversion. Yield reporting separates net deposits, withdrawals, share-price changes and fees. A seven-day observed annualization can be displayed when the history is sufficient; the observation window and calculation assumptions are visible. It is a historical estimate and can change or become negative.

Vault allowlisting considers implementation and upgrade rights, asset gates, rounding behavior, share inflation protection and the ability to enforce a minimum output. Deposits pause on unexplained conversion changes, unavailable valuations or curator events that breach the release's risk policy. Users retain a documented route to inspect and withdraw through the underlying protocol where available.

Tradient does not guarantee principal, fixed yield or instant redemption. It does not pool client positions into a Tradient-controlled investment fund.

12 Bridging and transfers

The bridge module presents routes into or out of Robinhood Chain. Routes are compared by net received amount, total fees, estimated time, failure handling and the contracts the user must approve. The user sees the actual asset and destination chain, not only a shared token symbol.

Robinhood's bridging documentation distinguishes the canonical bridge from third-party routes. Canonical withdrawals include a challenge period and a later claim step; fast routes have different liquidity and counterparty assumptions. Tradient displays route-specific timing and completion requirements instead of giving one universal bridge ETA. [10]

Bridge in

The source wallet authorizes the source-chain transaction. A Robinhood Chain session key does not gain authority on that source chain. Tradient binds the destination address to the user's selected account and tracks source submission, source confirmation, relay or claim state and destination receipt separately.

Bridge out and ordinary transfers

Destinations come from the owner-managed address book. An entry records address, chain, asset compatibility, ownership evidence where the route requires an own-wallet destination, activation time and policy version. A destination change restarts the 24-hour cooling-off period and invalidates pending quotes using the old entry.

Every transfer has a preview and single-use execution identifier. The preview contains recipient, network, asset address, amount and fees. The executor accepts no recipient override. Adding a recipient is unavailable to the AI tools. Verified self-withdrawal through the owner wallet remains possible independently of delegated transfer settings.

Failure and recovery

A successful source transaction does not establish bridge completion. The tracker retains provider reference, source hash, destination hash and refund or claim instructions. Failed relays and delayed claims enter an exception queue. Refunds return only to the originally authorized recovery address.

Retransmission follows provider-specific idempotency rules. The service never creates a new deposit because an existing transfer is merely delayed. Claim steps require separate consent where they create a new transaction or fee. Bridge risk remains visible until the destination asset is received or a refund has been reconciled.

Bridge adapters activate only after a complete small-value route succeeds in both directions, including an induced failure or refund test where the provider supports testing. Unsupported asset routes return a clear unavailable result instead of routing through an unreviewed provider.

13 Perpetual futures integration

Perps are optional, off by default and activated only after a separate derivatives release gate. The integration target is Lighter's Robinhood Chain instance. Robinhood documents this as a separate domain with its own execution and liquidity, not the same venue environment as Lighter Core. [11]

The adapter pins the venue API environment, signing domain, market identifiers, collateral assets and settlement contracts together. A chain ID used for a wallet transaction is not assumed to be the signing-domain identifier used by an external venue. The first supported collateral is an approved USDG deployment; additional collateral requires a separate release review.

Position controls

The default leverage ceiling is 2x. The owner may raise it to a product hard maximum of 5x through the dashboard after eligibility checks. New margin allocations are limited to 10% of the user's reconciled portfolio value by default. All open positions and pending orders count toward exposure checks.

Each opening order includes a defined stop-loss unless the owner explicitly changes that policy. A stop is an execution instruction subject to market conditions; it does not cap realized loss with certainty. The system monitors mark price, estimated liquidation price, collateral and funding information. An alert threshold is reached when absolute mark-to-liquidation distance divided by mark price is at or below 10%.

Separate deposit and trading authority

Onchain collateral deposits use wallet permission checks. Venue orders use a separately restricted API credential. Neither the credential nor wallet session key is exposed to the AI. Deposits credit only the user's identified venue account. Withdrawal destinations are constrained to the user's authorized wallet where supported; unsupported restrictions require owner confirmation or disable the action.

Opening, closing, collateral movement and order amendment all require previews and execution identities. A reduce-only close is validated against the current position so that retries cannot accidentally reverse direction. Partial fills update exposure before subsequent orders are accepted.

Activation requirements

The adapter must prove deposit attribution for the selected wallet type, credential creation and revocation, test order placement, partial fills, stop behavior, close and withdrawal. Country and account restrictions are evaluated separately from spot access. A venue API outage pauses new positions while retaining reconciliation and any available position-reducing controls. No perps access is inferred from a successful spot trade.

14 Centralized exchange connections

Exchange connections let eligible users operate existing accounts through their AI assistant. This balance remains in the exchange's custody. Tradient holds a restricted trading credential, so its service policies are not equivalent to onchain wallet restrictions.

The adapter layer uses CCXT where appropriate and exchange-specific APIs for capabilities the unified library does not represent precisely. Each venue requires its own production and sandbox validation; library support alone does not establish eligibility, contractual permission or equivalent order semantics. [12]

Venue selection for Western users

Exchange coverage follows the availability of each venue's regional entity, supported API products and permission controls. Coinbase, Kraken and eligible regional entities of other major exchanges are integration candidates, subject to those checks. No named venue is represented here as a completed Tradient integration or as universally available in Western markets.

Read access is activated before order placement. The owner creates a dedicated API key with required read and spot-trading permissions only, withdrawals and account transfers disabled, and fixed egress IP restrictions where available. Tradient rejects unsafe keys when permissions can be inspected. If the necessary restriction cannot be verified adequately, write access remains off.

Orders and budget enforcement

An order preview records venue, regional entity, account, pair, side, order type, size, limit price, fee estimate and policy version. The execute call consumes a single quote. The venue receives a persistent client-order ID where supported. Unknown outcomes are reconciled by ID, fills and account history before any retry can create another order.

Price and amount precision, minimum notional, supported order flags and fee currencies are venue-specific. Partially filled orders count toward budgets as fills occur while unfilled exposure remains reserved. Cancellation is not complete until the venue acknowledges it and fills are reconciled.

The platform exposes no exchange withdrawal or internal-transfer tool. Initial exchange support covers spot accounts only. Margin, borrowing, derivatives and subaccount administration are excluded unless separately specified and released.

Unlinking immediately disables Tradient requests, deletes the active encrypted credential reference and tells the user how to revoke the key at the venue. Deleting a local credential does not cancel an order already accepted by the exchange. Open orders are displayed with their cancellation status and remaining user action.

15 MCP authentication and interoperability

Tradient exposes a remote MCP service over Streamable HTTP. The MCP server is a protected resource; a separate authorization component issues user-bound tokens. The same backend services can also support a documented API for approved developer integrations.

The authorization design follows the current MCP authorization specification: protected-resource metadata, authorization-server discovery, resource-bound tokens and modern OAuth protections. Client ID metadata documents or pre-registration are preferred where supported. Dynamic client registration is retained for compatible clients that require it, rather than treated as a universal requirement. [8]

Connection model

Every connection records the user, client identity, approved scopes, creation time, last use and revocation state. Tokens are validated for issuer, audience, expiry and connection status on each request. PKCE, exact redirect matching and issuer validation protect authorization flows. Access tokens expire after one hour; refresh grants last at most 30 days and rotate when used.

Scopes are read, trade, transfer, earn, perps, automations and cex. Scope grants are intersected with product eligibility and wallet permissions. A cex grant does not authorize wallet trading. The tool catalog lists only enabled modules for that connection and environment, while the server validates every requested operation regardless of discovery output.

Tool contract

Inputs use schemas with explicit asset identity, decimal-string amounts and permitted enumerations. Outputs contain a short summary and structured data with source, timestamp, status and warnings. Error responses use code, message, retryable and fix fields. A quote includes expiry and the approval state, so a client can recover without inventing an action.

Read tools are marked read-only. Transaction tools carry appropriate destructive and idempotency annotations, but these annotations are guidance to clients; authorization is enforced on the server. No tool accepts owner keys, API secrets, arbitrary calldata or a replacement recipient at execution time.

Compatibility is certified by client version and mode. Release tests cover a browser-based assistant connector, a desktop or coding client and MCP Inspector. ChatGPT, Claude and Cursor are test targets where their connector capabilities are available. A client that cannot display or complete the required confirmation flow receives read-only access or a dashboard approval link.

Connected models can research and reason using Tradient data. They do not need direct access to wallet credentials, and tool access does not grant a bulk training-data license to Tradient's private user records.

16 Fees business model and token boundary

The initial spot-execution fee specification is 25 basis points, equal to 0.25% of the disclosed fee basis. The rate is displayed before execution and recorded with every quote. Any production fee change is published, versioned and shown in the next preview; it never changes an already approved quote.

0x supports integrator-fee parameters including swapFeeBps, swapFeeRecipient and swapFeeToken. Tradient uses the provider-returned fee amount and token as the accounting authority for the preview and then reconciles the actual fee from settlement. Route and fee-token rules are validated before release. [13]

Action	Tradient fee treatment
Confirmed spot swap	25 bps on the disclosed basis, collected once through the supported route
Failed or blocked swap	No Tradient execution fee; a submitted failed transaction can still incur network costs
Read-only portfolio and market tools	Included in the published access allowance
Deposit, transfer and bridge	No added Tradient execution fee in the initial schedule; external costs are disclosed
Earn and perps	No fee is enabled until a module-specific schedule is published and included in previews
Exchange orders	Exchange charges apply; no undisclosed fee is added to the user's venue account

For a purely illustrative $100 fee basis, 25 bps equals $0.25. Gas, spread, venue costs and bridge charges are separate. The preview identifies whether the fee is deducted from input or output and shows the actual maximum spend and minimum received amount. Sponsorship is subject to quotas and is not described as permanently free execution.

Subscription tiers can fund higher data quotas, richer analytics and more automation capacity. Prices and quotas are published before those tiers are sold. Paid access never weakens owner approvals, legal restrictions, contract checks or risk limits. Operating costs include RPC and data, signing, sponsored gas, indexing, hosting, support and security maintenance; gross fee revenue is not distributable profit.

$TRADIENT

The product identifier is $TRADIENT. A separate token launch schedule governs supply, allocation, vesting and any access benefits. No token sale, staking yield or revenue entitlement is established by this product whitepaper. Wallet ownership, emergency exit and recovery never require holding the token.

A separately released token schedule may define access allowances or fee discounts with explicit thresholds, measurement windows and fallback access. Until that schedule is published and implemented, no token benefit is represented as active. Tradient remains a usable service whose access and execution economics can operate without a token.

17 Security privacy and emergency controls

The threat model includes a manipulated AI, compromised OAuth connection, stolen delegated signer, malicious route, vulnerable wallet or protocol, stale data and incorrect reconciliation. Independent controls constrain these failures; no single label such as simulation or non-custodial establishes complete safety.

Signing boundary

The signing service is isolated from public request handling. It accepts only versioned execution records with a policy decision, reservation and approval reference. Secrets remain in managed signing or KMS infrastructure. Administrative credential access is separated from ordinary service access and audited. Session keys cannot install validators, change owners or upgrade the wallet.

Token metadata, documents and website text are untrusted data. Such content cannot grant permission, change tool schemas or substitute transaction destinations. The application decodes and validates transaction data independently of an AI explanation. Permission tests include nested calls, allowance misuse, malicious recipients and attempts to bypass cumulative budgets.

Emergency behavior

The kill switch immediately rejects new writes, invalidates connections, pauses workers and starts credential revocation. Onchain revocation becomes effective when the revocation transaction is confirmed. External venue keys and accepted orders require their own revocation or cancellation. The UI distinguishes local pause, revocation submitted and revocation confirmed.

Transactions already submitted may settle before revocation. Pending bridge transfers and open positions remain visible. Users receive independent owner-wallet exit instructions and exchange revocation instructions. A platform pause does not imply that every third-party order or financial exposure has disappeared.

Privacy and records

The MCP connection receives only the user's authorized account data. The platform does not need complete AI chat history to execute tools. Data minimization, field-level redaction and request-ID logging keep credentials out of model context and application logs. External model providers retain their own data-handling terms, which users review in their selected client.

A retention schedule separates active account records, append-only financial audit events and operational logs. Private records are deleted or restricted when no longer needed, subject to applicable recordkeeping obligations. Users receive export and deletion channels. Public blockchain records cannot be erased by Tradient.

Before real-money delegation, an independent review covers wallet permissions, route decoding, signer isolation, authorization, accounting and emergency behavior. Material findings block release. The public release record identifies reviewed versions and unresolved limitations without implying that an audit guarantees safety.

18 Operating standards and release gates

Feature activation is recorded in a public module-status register. A signed internal release manifest links provider versions, deployment addresses, test results, market access and fee configuration. Owners can see whether a capability is disabled, testnet-only, limited mainnet or generally available.

Measurable service targets

Cached reads target p95 below one second. Firm quotes target p95 below three seconds, and eligible signed submissions target p95 below five seconds after final approval. Trigger evaluation targets submission within ten seconds after receipt of a usable price update. These are engineering objectives, not guarantees about upstream freshness, chain inclusion or order fills.

Monthly MCP availability targets 99.5%. Independent RPC checks, provider timeouts, circuit breakers and documented failover support that goal. Failover cannot change an approved route without requoting. Reconciliation has priority over accepting new writes during an incident. Backups target a 15-minute database recovery point and a four-hour recovery time, validated through restoration exercises.

Release gate	Required evidence
Core testnet	Setup, data, quote, approval, swap, receipt and revoke work end to end. Duplicate calls, stale quotes, wrong users and out-of-budget requests are rejected.
Limited mainnet	Reviewed permissions and signer boundary; no unresolved critical or high security findings; approved launch-country matrix; team-funded round trips; support and incident runbooks.
Automations and extended modules	Restart-safe scheduling, accounting and revocation; each Earn, bridge or exchange adapter passes its own lifecycle and failure tests.
Perps	Venue-account and signing compatibility, country restrictions, partial fills, stops, liquidation alerts, close, withdrawal and independent key revocation are demonstrated.

Phased delivery

Phase 1 delivers the owner-confirmed testnet core, market data, notifications and dashboard. Bounded delegation is enabled only for action types with enforceable permissions. Phase 2 introduces limited mainnet, automations, approved Earn and bridging routes, and individually validated exchange connections. Phase 3 adds perps and optional richer client confirmations after their separate gates pass.

Dates are set from demonstrated integration progress rather than assumed provider support. Any unavailable adapter can remain disabled while the validated core operates. Expansion into a new country or product class requires its own eligibility and service review.

Tradient's administrators can change service configuration and pause adapters, but cannot broaden a user's signed wallet authority. Config changes are versioned and material changes trigger new previews or reauthorization. Token ownership does not confer authority to change user wallets or bypass the release gates.

Appendix A Core tool catalog

Tool names are stable interface contracts. Discovery is filtered by scopes and enabled modules. Decimal amounts are strings; assets resolve to chain and address before a preview. An execution response always includes action_id, state and the relevant chain or venue reference when available.

Tool	Inputs and result contract
get_portfolio	Optional account filter; returns reconciled holdings, valuation source, timestamp and known or unknown basis status
get_history	Cursor, type and time range; returns normalized events and evidence references
get_deposit_address	Selected enabled chain; returns owned address and explicit network identity
list_address_book	Returns activated and cooling-off recipients; never creates a recipient
preview_send	Recipient label, asset and amount; returns immutable preview and approval state
execute_send	quote_id only; returns existing or newly submitted action state
get_price	Resolved assets; returns price convention, source, age and market state
get_chart	Asset, pool, interval and range; returns time-series data and coverage limitations
search_token	Query; returns candidate addresses, verification status and ambiguity
token_info	Resolved asset; returns decimals, classification, liquidity and relevant metadata
check_token_safety	Resolved asset; returns observed checks, missing evidence and explained indicators
market_status	Asset or market; returns session, halt information and observation time
quote_trade	Side, asset, amount, amount_type, pay_with and slippage; returns quote_id, net economics, route and expiry
execute_trade	quote_id only; returns action_id, state, transaction reference and realized fill when confirmed
get_trade_status	action_id or transaction hash; returns submitted, reconciling, confirmed or failed state
explain_transaction	Transaction hash; returns decoded actions and actual transfers with source references

The service resolves symbols before building transaction data. A user-facing tool can accept a searched token address, but the executor never treats that input as permission to call arbitrary contracts. A new token can be inspected while remaining ineligible for trading.

Read methods use independent read quotas. Preview methods reserve no permanent trading authority. Execution methods use an atomic quote-consumption operation and are idempotent by action identity. The full set is organized into task-specific discovery profiles rather than exposed as an unrestricted catalog to every connected model.

Appendix A Automation and Earn tools

Creating an automation produces a draft mandate and approval link. The state becomes active only after owner authorization. Changing size, assets, schedule or expiry beyond the original mandate requires a new signed mandate version. Cancelling or pausing can only reduce authority.

Tool	Inputs and result contract
create_limit_order	Side, asset, size, reference threshold and expiry; returns draft mandate, estimated budget and approval requirement
create_dca	Asset, spend amount, funding asset, schedule, timezone and end condition; returns next occurrence and mandate preview
set_stop_loss	Asset or position, reference threshold and bounded quantity; returns draft stop mandate
set_take_profit	Asset or position, reference threshold and bounded quantity; returns draft take-profit mandate
create_rebalance	Target weights summing to 100%, drift tolerance, frequency and budget; returns proposed legs and authorization object
list_automations	Status and cursor; returns owner, policy version, budget, expiry and run states
pause_automation	automation_id; pauses future execution without claiming submitted orders are cancelled
cancel_automation	automation_id; cancels mandate and reports any external cancellation still pending
list_earn_options	Returns approved vault deployment, asset, observed yield window, limits and risk notes
get_earn_positions	Returns shares, asset-equivalent value, net flows and available withdrawal information
preview_earn_deposit	Vault and asset amount; returns minimum shares, costs and recipient
preview_earn_withdraw	Vault and asset amount or full exit; returns redeemable amount, liquidity and any queue
execute_earn	quote_id only; returns submitted action and reconciled share or asset changes
set_price_alert	Asset, comparison, threshold, window and channel; returns alert identity and source convention
list_alerts	Cursor; returns active conditions and delivery status
delete_alert	alert_id; stops future delivery for that alert
daily_briefing	Returns sourced portfolio change, cash flows, fees, positions and outstanding actions
set_briefing_schedule	Local time, timezone and channel; returns next scheduled delivery

Alert delivery is separate from financial execution. The notification service deduplicates each alert occurrence and reports channel failures without changing trading authority. A missed email does not cause a repeated trade. Briefings distinguish confirmed activity from pending or estimated figures.

Appendix A Perps bridge and control tools

Tool	Inputs and result contract
get_perp_markets	Optional market query; returns environment, contract specifications and current source timestamps
get_perp_account	Returns identified venue account, collateral, positions, liquidation estimates and reserved exposure
preview_perp_deposit	Approved collateral and amount; returns destination account, chain preview and crediting assumptions
preview_perp_withdraw	Asset, amount and approved own-wallet recipient; returns availability and relevant venue conditions
preview_open_position	Market, side, size, leverage, order type and protection levels; returns margin and execution preview
preview_close_position	Position and quantity or percentage; returns reduce-only close terms
execute_perp	quote_id only; returns deposit, withdrawal or venue order action state
cancel_perp_order	Existing order ID; returns cancel state and any fills already observed
quote_bridge	Direction, chains, asset, amount and saved destination; returns selected route, net output, timing and claim requirements
execute_bridge	quote_id only; returns authorized source action and tracking identity
get_bridge_status	Tracking identity; returns source, relay, destination, refund and claim states
get_limits	Returns effective policy, usage, credential expiry and enabled modules
tighten_limit	Rule and stricter value; applies a reduction and invalidates affected quotes
get_pending_approvals	Returns user-bound approvals with fresh-preview links and expiry
pause_agent	Connection or wallet scope; stops new actions and reports external exposures separately

Position closing uses preview_close_position plus execute_perp, preserving the same two-phase contract as other money-moving operations. Closing a position can incur fees and change exposure; it is not treated as an unreviewed administrative action.

The source-chain signer approves a bridge-in. execute_bridge cannot impersonate a source wallet; when that signature is outstanding it returns NEEDS_SOURCE_SIGNATURE with a dashboard link. A preview is never reported as a bridge submission.

Control methods cannot expand signing permissions. User instructions to raise a cap, enable leverage, add recipients, replace the owner or reactivate a paused wallet return USE_DASHBOARD. Authentication of a chat client does not establish owner presence.

Appendix A Exchange tools and response fields

Tool	Inputs and result contract
list_exchange_accounts	Returns venue, regional entity, permission verification and read or write availability
get_exchange_balances	Account; returns assets, free and reserved balances, valuation and observation time
get_exchange_price	Account and resolved pair; returns bid, ask, last and precision constraints
quote_exchange_order	Account, pair, side, type, amount and optional limit; returns quote_id, fees and reservation requirements
execute_exchange_order	quote_id only; returns persistent action and venue client-order identifiers
get_exchange_orders	Account, status and cursor; returns orders, fills and remaining reservations
cancel_exchange_order	Account and existing order ID; returns acknowledged or pending cancellation and observed fills

Common preview record

A preview includes quote_id, action_kind, user_id, connection_id, wallet_or_account_id, environment, resolved assets, integer-unit amounts, display amounts, source timestamps, fee breakdown, destination, route digest, minimum output where applicable, policy_version, approval_state and expires_at. Sensitive internal fields are withheld from client responses where they are not required for presentation.

Common execution record

An execution includes action_id, consumed quote, approval reference, mandate reference when relevant, budget reservation, signer credential reference, transaction or venue identifiers, state transitions and reconciliation timestamps. The model never receives private credentials. Confirmed results use actual fills or transfers and retain the original preview for comparison.

Standard errors

RULE_MAX_TRADE, RULE_DAILY_CAP, TOKEN_NOT_ALLOWED, REGION_BLOCKED, MARKET_HALTED, STALE_PRICE, PRICE_DEVIATION, QUOTE_EXPIRED, QUOTE_ALREADY_USED, POLICY_CHANGED, NEEDS_APPROVAL, NEEDS_SOURCE_SIGNATURE, NO_ROUTE, SIMULATION_FAILED, SIGNER_UNAVAILABLE and EXECUTION_RECONCILING have explicit recovery instructions.

Errors do not leak private account identifiers or infrastructure secrets. Retryable data failures tell clients when to retry. Authorization failures require the correct user action, not repeated execution attempts. A reconciling response includes the existing action ID so the AI can check status without creating another trade.

Appendix B Authoritative records

Postgres is the authority for identities, signed policies, quote use, mandate versions and reservations. Row-level access controls protect user-facing queries. Privileged services also scope every read and write by the authenticated user; a service credential is not permission to omit tenancy checks.

Record group	Minimum retained fields
Users and eligibility	User ID, verified account status, country evidence reference, timezone, enabled products and eligibility version
Wallets and owners	User, chain, account address, implementation version, owner validator and recovery configuration
Connections	User, client identity, scopes, token or grant references, last use and revocation state
Credentials	Public identity, KMS reference, purpose, hard permissions, expiry and confirmed revocation status
Policies	Version, limits, allowlists, effective time, owner approval evidence and superseded version
Quotes and approvals	Immutable terms, digest, expiry, status, approver evidence and consumed action
Actions and fills	User, origin, quote, mandate, account, nonces or order IDs, actual fills, fees and state history
Budget reservations	User, policy, action, asset and USD valuation evidence, reserved, consumed and released amounts
Mandates and occurrences	Signed definition, schedule, expiry, occurrence key, worker lease and result
Positions and basis	Asset identity, quantity, basis quality, methodology, source events and correction history
Recipients and notifications	Owner-approved addresses, cooldown, channels, delivery preferences and deduplication identity
Audit and release records	Request ID, connection, decisions, manifest version, source snapshots and incident references

Consistency rules

Transactions consume quotes, create actions and reserve budgets atomically. Unique constraints prevent duplicate quote execution and duplicate mandate occurrences. Signer requests are linked to durable actions before submission. Terminal failures release only unused reservations, and position updates are idempotent by fill or transfer identity.

System clocks use UTC; local time is a display and scheduling preference. All calculations record token decimals and price source versions. Database numeric types or integer-unit representations are used for monetary amounts; binary floating-point arithmetic is excluded from settlement and accounting logic.

Backups contain encrypted sensitive records and access is audited. Secret key material is not copied into the ordinary database backup. Restoration exercises verify both data recovery and credential isolation before write services are reopened.

Appendix C Acceptance scenarios

These scenarios are mandatory release evidence, not claims that the tests have already passed. Each adapter adds venue-specific cases to the common suite. Demonstrations retain the environment, deployment version, inputs, expected result and observed transaction or order references.

Scenario	Required outcome
Wrong user or client uses a quote	Rejected before reservation or signing; no private quote details leaked
Two clients execute one quote	Exactly one action identity; the second request returns its status
Simultaneous $80 actions with $100 remaining	One fits; the other is blocked by atomic budget reservation
$75 trade with default policy	Requires owner approval; $150 remains blocked by the $100 ceiling
Quote expires during approval	New terms are displayed and approved; the old quote cannot execute
RPC times out after submission	Existing transaction is reconciled; no second independent trade appears
Price feed is stale or multiplier changes	Writes pause or obtain corrected valuations; adjustment is not applied twice
Route changes recipient or nested call	Rejected independently of the AI's explanation and router name
Delegation lacks enforceable spending bounds	Delegated mode stays off; owner-confirmed execution remains the fallback
Session is revoked with an order pending	New writes stop; revocation and outstanding order states remain distinguishable
Worker crashes during a trigger	One occurrence resumes through its durable action record without duplicate submission
Deposit arrives with unknown acquisition cost	Balance is recognized; performance does not treat it as zero-cost profit
Vault cannot redeem the full position	Available amount and restriction are shown; no instant full withdrawal is promised
Exchange order partially fills then cancels	Fills remain in history; only remaining unfilled exposure is released
Perp close is retried after a partial fill	Reduce-only sizing cannot open a reverse position

Security review includes malicious token metadata, replayed OAuth callbacks, wrong token audiences, leaked credentials, concurrent budget bypass attempts and owner-recovery abuse. Operational review includes provider outages, notification failure, indexer lag, reorganization handling and restore-from-backup.

Public release notes state the exact modules, country scope and permission modes enabled. Any unsupported control remains a restriction on the feature rather than an unverified safety claim in marketing.

Appendix D References

Primary documentation was reviewed on 6 November 2026. References support network and interface facts; Tradient's limits, fees, architecture and release gates are the product rules defined in this whitepaper. Provider documentation, issuer restrictions and deployed behavior must be revalidated for each production release.

[1] Robinhood Chain. Connecting to Robinhood Chain. Network IDs, RPC endpoints and production infrastructure guidance.

https://docs.robinhood.com/chain/connecting/

[2] Robinhood Chain. Stock Tokens. Instrument description, issuer and geographic restrictions. The issuer prospectus and final terms govern applicable restrictions.

https://docs.robinhood.com/chain/stock-tokens/

https://docs.robinhood.com/rhj

[3] Robinhood Chain. About Robinhood Chain. Account abstraction and ecosystem infrastructure.

https://docs.robinhood.com/chain/

[4] ZeroDev. Permissions and Call Policy. Delegated credentials, target and function restrictions, parameter constraints and policy composition.

https://docs.zerodev.app/sdk/v5_3_x/permissions/intro

https://docs.zerodev.app/smart-accounts/permissions/policies/call

[5] 0x. Supported Chains and Providers. Robinhood Chain Swap API support.

https://docs.0x.org/docs/introduction/supported-chains

[6] 0x. RWA Quickstart. Team opt-in, indicative and executable quotes, decimals and allowance flow.

https://docs.0x.org/rwa/rwa/guides/quickstart

[7] Robinhood Chain. Stock Token APIs. Metadata, price conventions, corporate-action multipliers, timestamps and halt information.

https://docs.robinhood.com/chain/stock-token-apis/

Appendix D References continued

[8] Model Context Protocol. Authorization specification, revision 2026-07-28. Resource server roles, metadata, client registration, token audience and authorization flow.

https://modelcontextprotocol.io/specification/2026-07-28/basic/authorization

[9] Morpho. Depositing and Withdrawing from Vaults. ERC-4626 integration and version-specific limits. The selected deployed vault's version takes precedence over generic assumptions.

https://docs.morpho.org/developers/earn/tutorials/assets-flow/

[10] Robinhood Chain. Bridging. Canonical and third-party route distinctions, withdrawal challenge periods and claim requirements.

https://docs.robinhood.com/chain/bridging/

[11] Robinhood Chain. Lighter Domains. Separate Robinhood domain, contract environment and deposit and withdrawal interfaces.

https://docs.robinhood.com/chain/lighter-domains/

[12] CCXT. Manual. Exchange-specific authentication, precision, orders, rate limits and sandbox behavior.

https://docs.ccxt.com/docs/manual

[13] 0x. Monetize Your App on EVM. Integrator fee parameters and fee-response handling.

https://docs.0x.org/evm/0x-swap-api/guides/monetize-your-app-using-swap

Definitions

MCP means Model Context Protocol, the interface through which compatible AI clients discover and invoke tools. A UserOperation is an account-abstraction execution request processed through a bundler. A session key is a delegated credential with defined permissions and expiry. A quote is an immutable preview whose identifier authorizes no changes to its terms. A mandate is an owner-approved standing instruction with bounded authority.

Principal means the amount being traded, transferred or allocated before separately itemized fees. A reservation prevents concurrent actions from spending the same policy budget. A receipt is the reconciled evidence of submission or settlement, not a prediction of execution. A release gate is an evidence requirement that must pass before a capability is activated.

The product specification does not establish a relationship with, endorsement by or authorization from any named AI client, wallet, issuer, venue or infrastructure provider. Those relationships and integration rights are assessed separately before distribution.