

RUNCHECK: End-user operating manual

Create a RUNCHECK agent with an automatic model API check, or verify an external agent.
Use your wallet to launch and operate.

Website: runcheck.bond · Edition: 18 October 2026 (Europe/Rome) · Audience: agent owners, token creators and testers.

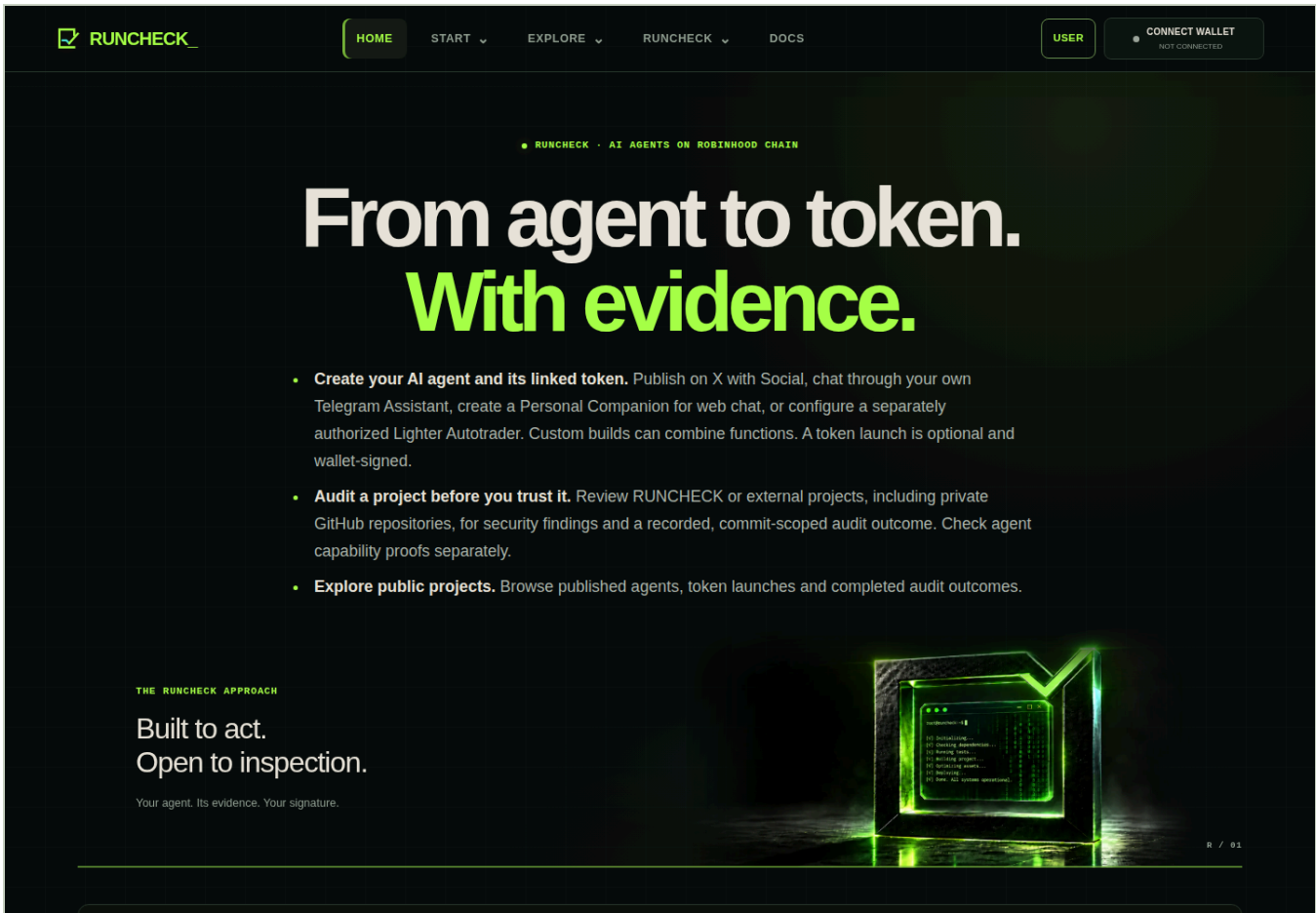
The website publishes only this English PDF edition. English and Italian Markdown sources remain internal; the Italian PDF is also internal. None of these internal files is offered as a website download. Official community links are in the footer: [X](#) and [Telegram](#).

Sections 1–9 introduce the product and its first-use tour. Sections 10–21 contain the detailed procedures, field reference and completion checks. Start with section 22 for practical use cases; section 23 answers everyday questions. This edition separates the two Home starting points, the guided Agent & Token Launchpad, token launch, the public token and agent storefronts, the public audit catalog and private USER lists. It retains all operational recipes, the automatic factory model check and bounded repository reviews: GitHub App onboarding, expanded dependency/import evidence, the pinned local/offline RUNCHECK RepoScan Static pass, separately consented RUNCHECK RepoScan Intelligence and trusted-host runtime evidence. Button names match the English interface. Examples are illustrative inputs, not recommended spending amounts or evidence of completed trades.

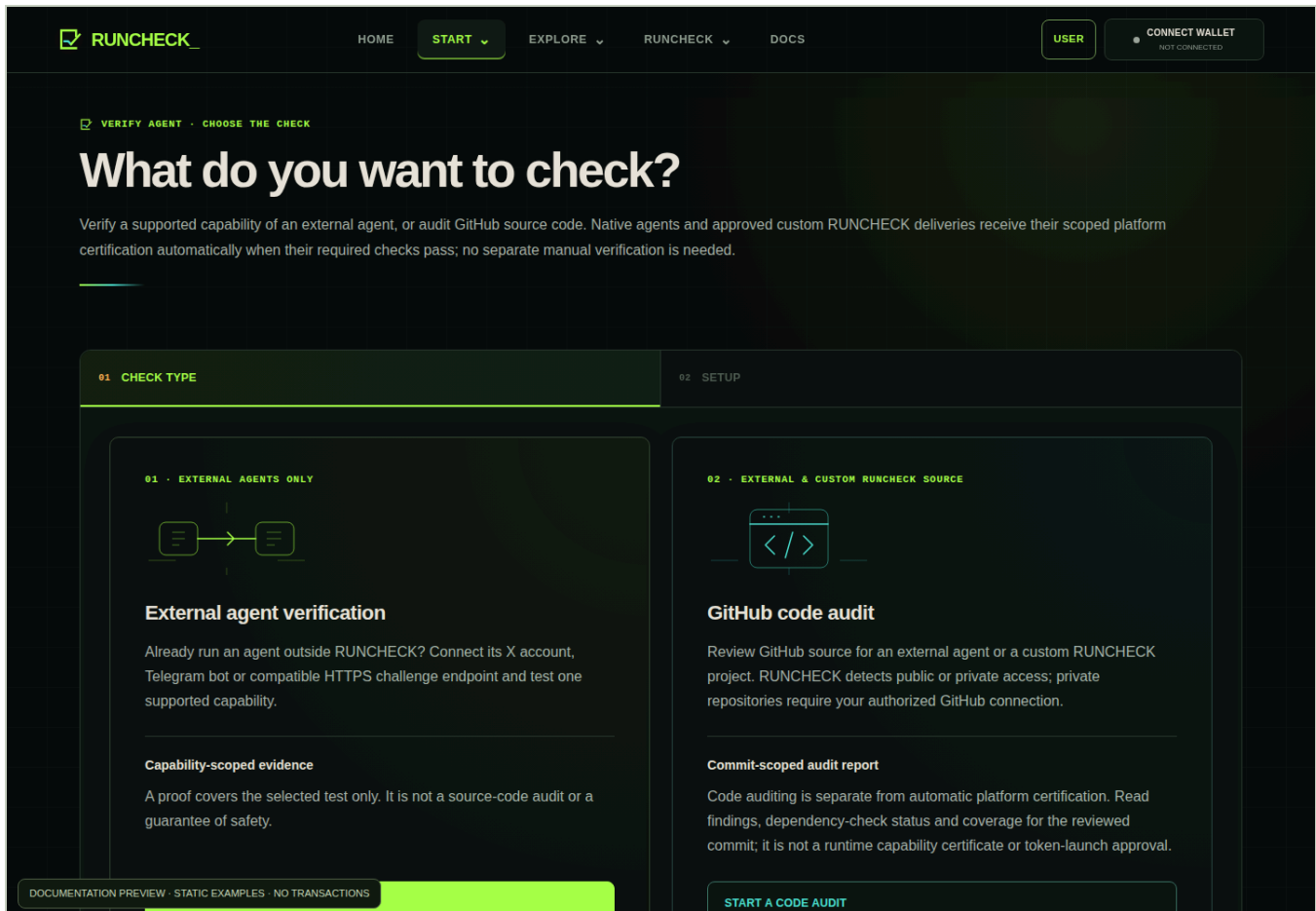
Task	Detailed procedure
Choose a use case and understand the result	22. Practical use cases
Navigate on a phone and find products, USER and help	1. Start here
Set up a wallet, find your accounts and browse the directories	10. Wallet and account preparation
Create the four agent types with your own model key	11. BYOK deployment recipes
Buy, confirm, replenish or restart hosted services	12. Managed AI and Managed X checkout
Inspect the automatic factory check or verify an external agent; publish a profile	13. Proof and publication recipes
Request a custom build and follow approval/publication	Custom agent requests
Audit an existing agent's public/private GitHub repository	Code audit for any agent
Chat, use voice, apply a trade proposal and trade manually	14. Runcheck Agent and manual order controls
Configure, activate and monitor a real strategy	15. Autotrader field reference and recovery
Manage capital and session permissions	16. Vault recipes and permission boundaries
Launch a token and perform all ten Pons V2 operations	17. Token launch and operation recipes
Publish and schedule on X	18. X publishing and scheduling reference
Set up a private bot or link the shared Runcheck Agent	19. Telegram setup and identity
Retire, revoke, archive and recover	20. Lifecycle and troubleshooting
Verify you have explored every public product path	21. Feature completion checklist
Understand costs, permissions, terms and daily checks	23. Everyday reference and FAQ

Navigation screenshots show the current routes. Retained control examples use labelled demonstration identities; their fields remain relevant even when the layout changes. These are navigation aids, not proof of real payments, funded balances or successful trades. No example address is a funding destination.

1. Start here



Home: three clear platform capabilities, direct public project storefronts, two focused entry points and the public Runcheck Agent. Local preview only; no signed wallet or transaction.



Current Verify Agent page: external capability verification or a separate GitHub code audit. RUNCHECK agent certification is automatic. Local preview, no completed operation.

RUNCHECK combines agent creation, capability verification, token launches, agent vaults, market chat and trading. Creating an agent, publishing its profile and launching its token are separate actions. You do not need a token to use an agent.

Home introduces three paths: create an AI agent and its linked token, audit RUNCHECK or external projects including private GitHub repositories, and explore published agents, recorded token launches and completed audits. Capability proofs are separate from code audits. Public market research with Runcheck Agent is also accessible from Home and the RUNCHECK menu. Code-audit outcomes describe the checked commit, findings and coverage; they do not guarantee that a project is safe.

Find your way around

Home has two entry points: **Agent & Token Launchpad** → **Verify Agent**. Tokens remains in the public-project section and header. Launchpad contains standard templates and **REQUEST A CUSTOM AGENT**. Verify Agent offers **VERIFY AN EXTERNAL AGENT** at /verify/custom for supported external capability challenges. Native agents and approved custom RUNCHECK deliveries receive scoped platform certification automatically when their required checks pass; do not register or verify them again here. Inspect their saved proofs in **USER** → **MY AGENTS**. **START A CODE AUDIT** opens the separate GitHub form at /review/new for external agents or custom RUNCHECK projects. Public/private access is detected automatically; private access requires authorization. Capability proofs and commit-scoped code reports are distinct evidence, not interchangeable safety guarantees.

The **Agent & Token Launchpad** card previews **Create agent** → **Automatic AI check** → **Review + sign token**. The AI response check runs automatically; token deployment does not. After **MODEL API VERIFIED**, click **CONTINUE TO TOKEN LAUNCH**: /launch?agent=AGENT_ID opens with your agent already selected. Complete the token settings, review the simulated Pons V2 transaction and confirm it in your wallet. You may stop at the agent without launching a token. Custom requests continue only after approval and eligible delivery.

Below the two starting points, **Explore what's already here.** links recorded token launches, published agents and public code-audit outcomes. **Start with a conversation.** opens the public Runcheck Agent with an illustrative question, not a pre-sent message or your private agent. **Know your next move.** opens Docs and the English user guide. These sections browse or explain existing features; they do not create agents, certify projects or send transactions.

Screen	What you can do
<u>Home</u>	Read the brief introduction, choose Agent & Token Launchpad or Verify Agent and open the public Runcheck Agent, RUNCHECK token launches or agent directory. Home has no creation form or embedded account dashboard.
<u>Agent & Token Launchpad</u>	Create a Personal Companion, Autotrader, Social X or Telegram Assistant, then follow one guided setup stage at a time. None is required to use the shared Runcheck Agent. External capability verification is a separate path.
<u>Token launch</u>	Select an eligible existing agent, complete the Launch Gate and review/sign its token transaction. Agent creation lives in Agent & Token Launchpad, not below this form.
<u>Custom Agent</u>	Request a custom agent or special integration. Follow your own approval and delivery in USER → My custom requests .
<u>Verify Agent</u>	Verify one supported capability of an external agent, or request a separate GitHub code audit for external or custom RUNCHECK source. Native/custom RUNCHECK platform certification is automatic, not a manual option here.
<u>Agent Audit</u>	Browse public reports and actual outcomes. VERIFY AN AGENT opens the check chooser at /verify; choose START A CODE AUDIT to reach /review/new; saved private requests live in USER → MY CODE AUDITS .
<u>Runcheck Agent</u>	Use RUNCHECK's shared research and manual-trading workspace without creating an agent. Public chat needs no wallet; personal account context, manual trading and the optional shared Telegram bot require their own sign-in/authorization.
<u>Agents</u>	Search explicitly published profiles and filter all, token-linked, currently verified or saved-live agents. These are saved public states, not a claim that the agent was natively created by RUNCHECK or checked live during this visit.
<u>Tokens</u>	Browse RUNCHECK token projects by default, search tokens and compare available price, trading activity and curve/pool details. ALL RUNCHECK combines Native RUNCHECK and External RUNCHECK. Open ALL PONS for the complete source market.
<u>USER</u>	Open a private list, then one item and its operation page. Public visitors cannot see drafts, request briefs, full private-repository reports or account controls; completed audit summaries are public automatically.
<u>Docs</u>	Resources groups User guide, Checks & evidence, Your account and Protocol & whitepaper. Legal links Terms of service, Privacy policy and Token launch terms directly, without an intermediate Legal page. No wallet is required.
<u>User guide</u>	Open help and these manuals. Whitepaper and terms are in the site footer.

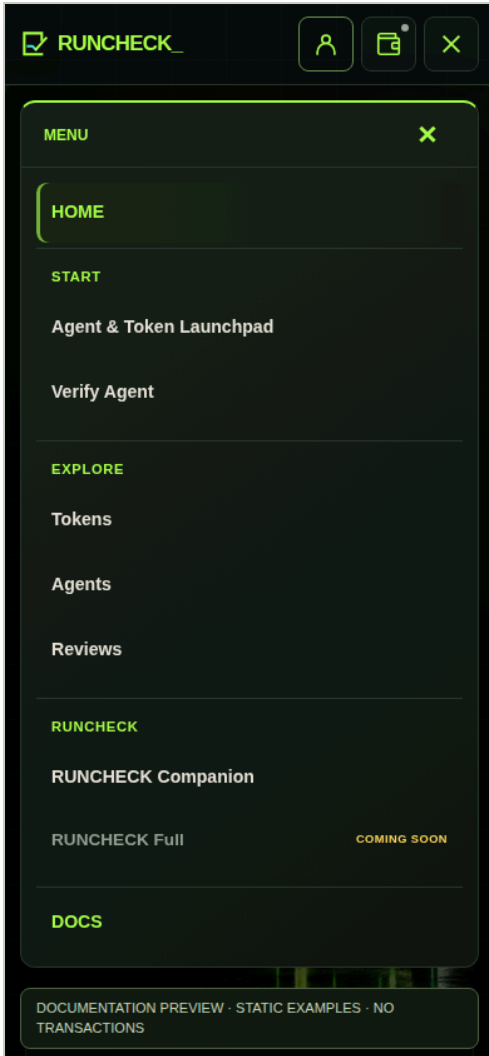
Use **CONNECT WALLET** in the always-visible site header to open one wallet popup. Choose your browser wallet or **Safe / contract wallet** there; opening the popup alone requests no signature. Safe remains available inside the same popup when already connected, not as a separate header button. On mobile, the control is a wallet icon with an accessible connection label and status dot. Sign-in proves ownership; it is not a payment and requires no agent. On desktop, the connected address appears in that same button across pages. Robinhood Chain (chain ID **4663**) is required for Robinhood transactions; **CONNECTED · CHECK NETWORK** means you are signed in but the correct transaction network has not been confirmed. A later transaction or trading delegation is a separate approval. Use the same owner wallet when returning to an agent.

The main navigation is **HOME**, **START**, **EXPLORE**, **FEED**, **RUNCHECK** and **DOCS**. **START** opens **Agent & Token Launchpad** or **Verify Agent**; **EXPLORE** opens **Tokens**, **Agents** or **Reviews**. **FEED** opens the public social feed at /feed. **RUNCHECK** opens **Runcheck Agent** at /full. **DOCS** opens guides and protocol resources. On desktop, click a group to open its dropdown; Escape closes it, and keyboard users can use Tab or arrow up/down from its trigger. These browsing destinations do not require a wallet. Account controls remain **ADMIN** → **USER** → **wallet**; **ADMIN** appears only after an authorized wallet signs in. Help and terms are also linked in the footer.

EXPLORE → **TOKENS** opens /tokens?filter=runcheck with **ALL RUNCHECK** selected, on desktop and mobile. Choose **ALL PONS** for the complete source market, **NATIVE RUNCHECK** for finalized wallet launches and approved custom **RUNCHECK** deliveries such as Polar, or **EXTERNAL RUNCHECK** for existing Pons tokens explicitly linked to a public external agent with a current passed repository audit and a current token-bound capability certificate. Background indexing is independent of the selected view; full-market catalog reads use the existing bounded cache when requested.

Mobile navigation: a short menu, direct pages

1. Tap the **three-line menu icon** (accessible name: **Open site menu**). The menu opens downward, leaving part of the page visible below.
2. Choose **HOME**, **FEED** or **DOCS** directly, or a destination under **START**, **EXPLORE** or **RUNCHECK**. The grouped mobile menu shows the same working destinations as desktop. **Runcheck Agent** opens the shared research and trading workspace. A destination opens its own page and closes the menu. **DOCS** is the only documentation entry in the menu: open **User guide** inside that hub. If the list is taller than the screen, scroll inside the menu to reach **DOCS**.
3. Use the header icons: **ADMIN** (shield, only for an authorized signed wallet), **USER** (person), then wallet. Account controls are not duplicated in the drawer. The **USER** overview uses category cards; the compact category strip appears only on its subpages.
4. Tap the page below the menu, use the close control or press Escape to dismiss it. Help, manuals, Whitepaper and terms are in the footer.
5. In **Runcheck Agent**, switch between **CHAT** and **TOOLS** when chat and tools cannot fit side by side. **TOOLS** contains **Markets**, **Vaults & auto** and **Telegram**. In **Agent & Token Launchpad**, use the current stage and next/back controls; token launch is a separate page, not a lower section of Home or the wizard.



Mobile navigation: grouped START, EXPLORE and RUNCHECK destinations, DOCS and centralized wallet controls. Public navigation only; no wallet connected.

TOKENS, AGENTS and REVIEW are reached through EXPLORE in the main header, without a second catalog navigation bar. From any page, **START** → **Agent & Token Launchpad** opens agent creation directly at `/launchpad`: choose a template, complete setup and the automatic model API check, then optionally review and sign its token launch. Page introductions show full-width titles and descriptions without shortcut buttons; operational controls remain in their content sections. Existing ready agents still use `/launch?agent=AGENT_ID`. Screenshots from the previous edition may show older header shortcuts; follow the current navigation described here. DOCS opens guides at `/learn`. Rounded cards improve navigation; they do not establish launch origin, certification or live health.

Your private lists and one operation at a time

Open USER → **My overview**, then select a card under **My projects** or **My account**. USER contains only your resources, not generic creation or public showcase shortcuts. Create through START; browse through EXPLORE. Lists open one item for its details. Overview does not load project or billing data.

Refresh is a compact secondary control beside its related data or at the end of the section. It repeats the named read only; it does not launch an audit, grant consent or sign a transaction. My token launches retains saved-attempt continuation; start a new agent/token from START, not from this history.

USER card	Dedicated page	Use it for
My agents	<code>/user/agents</code>	Select an owned agent and open its workspace.
My token launches	<code>/user/launches</code>	Inspect saved launches and recover the exact pending transaction.

USER card	Dedicated page	Use it for
My custom requests	/user/custom	Follow requested → approved → published custom deliveries.
My code audits	/user/audits	Open your saved public/private repository requests and reports.
My credits & services	/user/services	Read confirmed AI credit purchases and saved entitlements; explicitly check one agent's available balance. BYOK spending is separate.
My GitHub connection	/user/github	Inspect or revoke wallet-linked GitHub App access; connect or reconnect during a private audit request.
My activity	/user/activity	Read saved control history, not exchange execution evidence.

In MY AGENTS choose **OPEN AGENT WORKSPACE**. The route carries the owned agent ID, for example /workspace?agent=AGENT_ID&view=overview; AGENT_ID is a placeholder, not an address. Use **OVERVIEW**, **RUNTIME**, **PROOFS**, **VAULT**, **AUTOTRADER** (trader agents only), **AI + SERVICES** or **ACTIVITY**. Their view values are overview, runtime, proofs, vault, autotrader, services and activity. Each view displays one operation area. Manual Lighter orders and 0x swaps remain in /full, separate from the agent's Autotrader and on-chain vault.

For Phantom on a phone, open the site in Phantom's in-app browser; ordinary mobile Safari/Chrome is not its connection path. See the [official connection instructions](#) and section 10.

Keep these balances separate

Balance	Purpose
ETH in your Robinhood Chain wallet	Network fees, ETH-paired launches and ETH purchases.
USDG in that wallet	Managed AI / Managed X payments; USDG-paired operations when selected.
An agent's on-chain vault	Capital held by that particular vault, under its owner and session rules.
A dedicated Lighter account	Collateral for perpetual trading. Funding your wallet or vault does not automatically fund Lighter.

Keep enough ETH for network fees even when paying in USDG. Follow the current screen quote; do not infer a fee from this guide. Never paste a wallet seed phrase or owner private key into RUNCHECK. Provider API keys belong only in the matching secure setup fields, not in chat.

Status labels: LIVE means the service is available. CONDITIONAL means you still need an identity, payment, balance, proof or signature. A status badge does not prove that your own transaction or trade succeeded.

Before signing: public token/configuration snapshots can be 10–15 seconds old. Transaction preparation performs fresh checks. If public data is busy, wait five seconds and use the retry control; never bypass an unavailable review.



2. Create and manage an agent

Open [Agent & Token Launchpad](#), then:

There are only two top-level steps: **AGENT TYPE** and **SETUP**. There is no Goal step or verification branch inside creation. Native and approved custom RUNCHECK agents obtain their scoped platform certification automatically after the required checks pass. **Verify Agent** is for external manual capability tests or separate GitHub audits of external/custom RUNCHECK source.

Choose by task and channel

Native template	What it does and where	Example / boundary
Social X	Publishes posts, replies and quotes through your authorized X account, with your content instructions, schedule and limits.	Share a project update on X or reply to a relevant post. Account authorization and publishing controls are required; creating the agent does not start posting.
Telegram Assistant	Your AI assistant answers you through your own Telegram bot in its one authorized private chat.	Message your bot: "Explain what a liquidity pool is." It answers you, not another agent. It is not a group moderator or an automatic alert service.
Personal Companion	Your own browser-based AI chat with the model, language and instructions configured for that agent. Open its own Companion link.	Ask your assistant to explain a market concept or available data. This template does not include Runcheck Agent's manual-trading panels or shared Telegram order updates.
Lighter Perpetual Autotrader	Runs an explicitly authorized perpetual-trading strategy on a dedicated Robinhood Lighter account. Configure and monitor it in that agent's workspace.	Use a supported DCA, momentum or mean-reversion strategy with reviewed limits. It needs funded collateral, its own API key and signed delegation; creation and token launch do not activate trading.

The agent type describes its task and access channel, not a different AI provider. A custom RUNCHECK delivery can combine channels and extra functions under its agreed scope; selecting a native template does not automatically connect it to other agents. **Runcheck Agent** at /full is a shared platform tool: no Personal Companion, Telegram Assistant or other agent creation is required. Its optional Telegram connection uses a bot supplied by RUNCHECK, not a BotFather bot you create. Section 19 separates that service from your own Telegram Assistant and public platform announcements.

1. Creation is the default path. Choose Social X, Telegram Assistant, Personal Companion or Autotrader on the type page (step=types); choosing one opens setup (step=setup). Choose **REQUEST A CUSTOM AGENT** on the same page for bespoke development with agreed scope and delivery. Name and description start empty; examples are placeholders or explicit copy buttons, not saved content. Select personality, language and any template-specific settings shown.
2. Connect and verify the owner wallet. Save the draft in the details stage (stage=details); successful saving advances to model/service setup (stage=model). The saved agent appears in MY AGENTS. If saving fails, the error and focus stay with registration. A reserved name/slug requires a different unique name; saving repeatedly does not free an existing address.
3. Choose **USE RUNCHECK AI** for hosted inference or **USE MY OWN AI KEY** for your provider account. Availability depends on the selected template.
4. Supply only the model, endpoint and credentials requested by that deployment path. Social and Telegram require their own account/bot setup as well.
5. Review and deploy. Wait until the exact runtime revision is ready. A saved draft is not a running agent.
6. In AI + deployment (stage=model), wait for **MODEL API VERIFIED**, then **CONTINUE TO TOKEN LAUNCH** opens /launch for this agent. RUNCHECK performs a real model-response check automatically; you do not start a runtime, trading, X or Telegram proof. If the check fails, correct the indicated provider problem and use **RETRY MODEL API CHECK** after one minute without buying or deploying again. MY AGENTS and TOKEN LAUNCHES preserve setup and launch history. The URL records the current stage; back/next navigation does not silently create another draft or authorize payment.

Description example: "Explain BTC and ETH market data in plain English and summarize portfolio risks without executing trades." Use **USE THIS DESCRIPTION** for a template-specific starting point. Describe the purpose, audience and result, within 180 characters. **Character example:** "Be patient and concise. Separate facts from

assumptions, ask for missing context and never claim an action succeeded without confirmation.” Use **USE THIS CHARACTER**, then edit the custom instructions. Character changes tone, not trading permissions or risk limits.

Pay for hosted services

Select **USE RUNCHECK AI** to see USDG credit packs, checkout, usage and top-ups. These panels stay hidden when no AI plan is selected or when **USE MY OWN AI KEY** (BYOK) is selected. Request a current quote, review the pack and payment destination, sign the requested wallet payment, then confirm it in the checkout. Wait for confirmed credit/entitlement before deployment or further use. Review remaining credits, usage and spending limits; top up through a fresh quote.

Before creating an agent, open [AI credit pack, price and limits](#). The same live catalog appears in the Personal Companion/Autotrader setup after selecting **USE RUNCHECK AI**. There is **one standard pack**, purchased one at a time; repeat a confirmed top-up to add another pack. One credit pays for one completed model call, not one token or necessarily one chat message. Credits have no time-based expiry; buying more does not increase the daily billing-account call limit. If the catalog is unavailable, no price is assumed. RUNCHECK manages provider access; this does not mean RUNCHECK owns the underlying AI model.

Managed X is a separate, time-limited service tied to the exact connected X account. It needs its own subscription, posting consent and available quota. Buying AI credits does not automatically buy X publishing. BYOK uses your own provider credentials and is not a free hosted subscription.

If a payment is pending, keep its transaction hash and use confirmation/recovery. Do not pay again just because the first confirmation is delayed.

Maintain or retire an agent

Action	Use it when / expected result
Open/manage agent	Continue setup, inspect health, edit supported settings or refresh state.
Redeploy	Apply a supported new runtime configuration; wait for the new automatic model API check.
Revoke/rotate credentials	Stop using an old provider connection. Retire the active generation first when the UI requires it.
Retire runtime	Stop that deployment and revoke its runtime access. Follow cleanup status and retry failed cleanup.
Archive test agent	Hide an eligible clean draft or retired test agent. Enter the displayed confirmation phrase. Current proof or an active launch can prevent archiving.
Recover archived vault funds	Use the archived-agent recovery area; archiving does not erase the on-chain vault.

Before retirement, stop delegated trading, resolve positions/orders and recover funds as needed. Retirement or archiving is not a withdrawal and does not erase billing, proof or transaction history.

Advanced Settings exposes connector references and a manifest control check. A manifest check records configuration; it is **not** a passing runtime challenge or permission to trade.



3. Prove, publish and launch

Verify an external agent capability

Native agents and approved custom RUNCHECK deliveries already receive scoped platform certification automatically when the required deployment and model-response checks pass. This proves the recorded scope, not trading performance or social publication. Inspect owned RUNCHECK agents and their saved proofs in **USER**

→ **MY AGENTS**, not in external verification. Viewing evidence does not run a new model check; if the model proof is missing or stale, open the agent's AI + deployment setup and explicitly retry the indicated check. [Verify Agent](#) → **VERIFY AN EXTERNAL AGENT** is only for the external challenges below. GitHub code review is separate and available for external agents and custom RUNCHECK projects.

1. Open Verify Agent → **VERIFY AN EXTERNAL AGENT** (/verify/custom) and select one supported capability: Companion Response, Paper Trade Proposal, Publishes on X or Replies on Telegram.
2. Complete its displayed requirements. An external runtime needs its real challenge endpoint; X needs the expected author identity and actual post evidence; Telegram needs its verified bot/private-chat setup.
3. Start the challenge and follow the exact instructions before it expires. Supply the requested evidence where applicable, then verify the result.
4. Open the resulting certificate. Check the capability, agent, runtime revision, timestamps and evidence. A configuration change or expired/revoked proof can require another challenge.

One passed capability proves that specific behavior. A paper-trade certificate is not live-trading authorization, profitability evidence or proof of every agent feature.

Existing external agent: choose [Verify Agent](#) → **VERIFY AN EXTERNAL AGENT**, then register its identity and actual verification endpoint/effect. This path supports proof, but an external-only deployment does not qualify for the RUNCHECK factory token-launch path.

Publish a profile

Open USER → **MY AGENTS** → the agent's **AI + SERVICES** view → its AI + deployment setup (/launchpad?agent=ID&intent=create&template=TYPE&step=setup&stage=model). In **PUBLIC DIRECTORY**, publish the ready agent and open its public profile. Check the displayed name, description and proof links. Unpublish to remove the directory listing; this does not stop the runtime or undo a token launch. Search it in Agents and inspect the certificate from a signed-out window.

Newly public agents and confirmed token launches can generate a branded image and a copy-ready announcement in the RUNCHECK Telegram alert group. This applies to standard, custom and external public profiles. Announcements use public project metadata and link to the accessible project page; drafts, private profiles, pending transactions and private request details are excluded. An existing token newly associated with RUNCHECK is labelled as linked, not as newly created. This feature does not post to X or activate a Social agent's publishing permissions. Removing the profile does not erase announcements or copies already shared elsewhere.

Launch a token with Pons V2

1. Open [Token launch](#). If no eligible agent exists, **START AGENT SETUP** opens Agent & Token Launchpad; complete creation there and return. Otherwise choose **SELECT AN AI AGENT · ALREADY CREATED**. In **LAUNCH GATE**, resolve the owner, automatic model API check, current deployment, acceptance and Pons readiness checks. This page contains only token setup and transaction review, not another agent wizard.
2. Enter the token name, symbol, description and artwork/social links. Token text fields start empty; **USE THIS AGENT'S DETAILS** copies the selected agent only when you ask. The connected creator wallet is the initial fee recipient; verify it in the review. Use a clearly marked test name during testing.
3. Review the quote asset: select **ETH** for an ETH pair. USDG is a separate supported choice when the screen offers it. Set the initial buy, creator tax and buyback preference.
4. Review current launch fees, initial-buy amount, predicted token address, economics and expiry. For an ERC20 initial buy, complete any exact approval, then prepare again.
5. Let RUNCHECK verify the deployment and simulate the exact transaction. Check the network, destination and total value; then confirm in your wallet.
6. Save the hash. Wait for reconciliation/finality, then inspect the linked token, curve/pool and agent certificate. Use pending-launch recovery after a reload; do not submit a duplicate launch.

No deployment verification or failed simulation means no signature should be requested. Refresh and investigate the displayed error. Token creation, approval and purchase are distinct transactions where the UI says so.

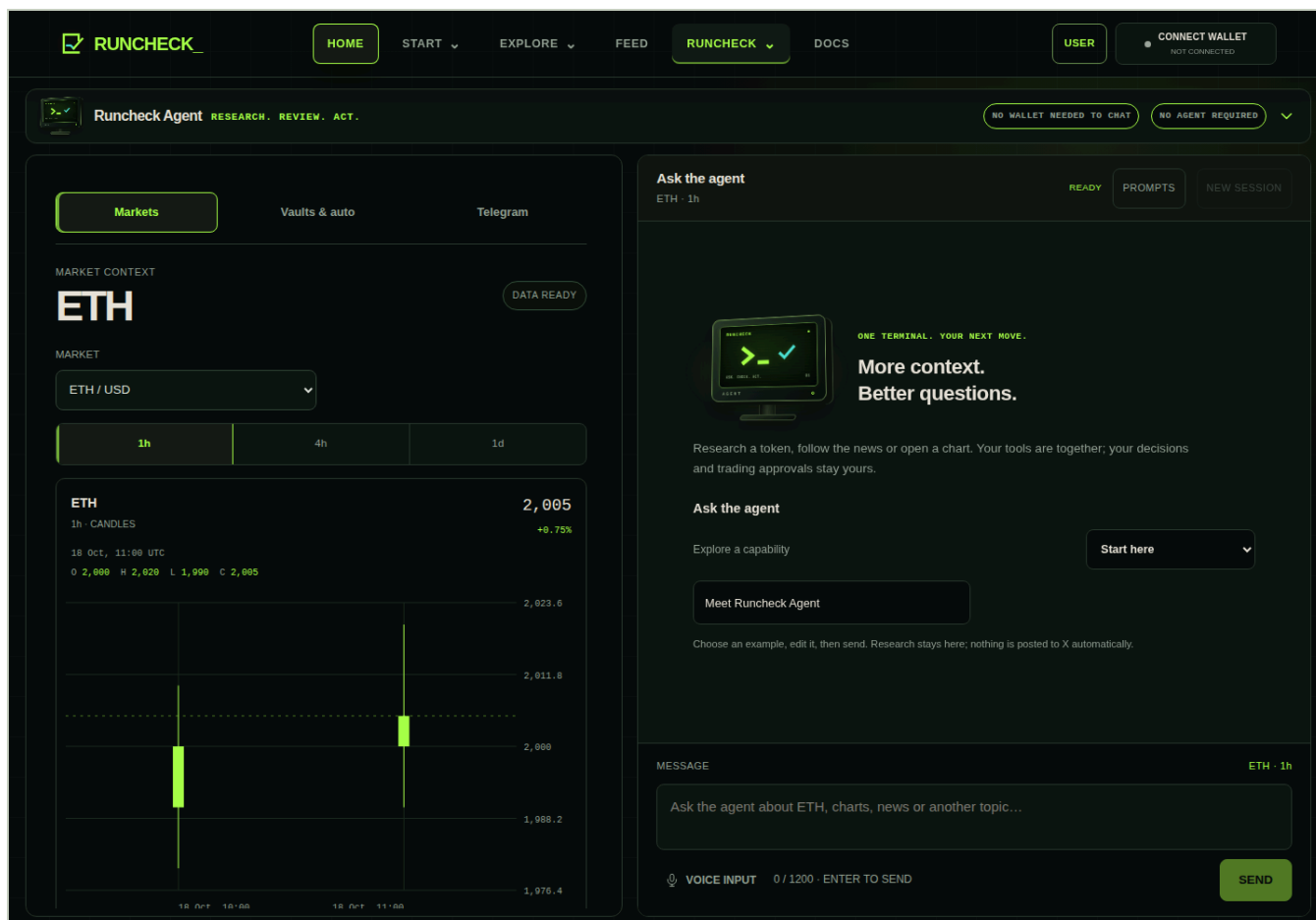
4. Use Runcheck Agent and manual trading

The RUNCHECK menu places **Trade history & PnL** directly below **Runcheck Agent**. It opens `/full/history`: sign in to review only your own available trade records and reported results. Hourly AI decisions remain in the app and may be HOLD rather than trades. **Share on X** opens an editable draft only after you choose it; nothing is posted automatically.

This is a source-scoped private journal, not a complete all-time account statement. Automation retains up to 100 completed execution records and 30 completed AI cycles plus the current cycle. The optional manual-account fill read returns up to 100 fills per page using your existing browser signer; choose **Load older fills** to continue through the history available from Lighter. The saved spot journal contains the latest 20 intents. A proposal or acknowledgement is not a confirmed fill. Reported realized PnL, unrealized PnL and funding remain separate; missing values are not zero and balances are not profit. The public X draft omits wallet, account, size, balances and PnL.

Runcheck Agent is the shared platform workspace at `/full`, not an agent you must create or deploy. Its public research chat, browser manual-trading panels and optional RUNCHECK-provided Telegram connection are separate from the **Personal Companion** native template. No personal agent, own BotFather bot or model API key is required to open this workspace or link its shared bot. Wallet sign-in, account funding and trade authorization remain separate requirements for the operations that need them.

The legacy `/companion` address remains supported for existing links; `/full` is the current public entry point. Personal Companion agents keep their own separate chat links.



Runcheck Agent: terminal avatar, public research conversation and Markets tools. Illustrative preview with sample chart data; no wallet connected, funded account or submitted trade.

The compact **Runcheck Agent · RESEARCH. REVIEW. ACT.** bar expands to explain the workspace. Select it again to collapse the description; the conversation and draft stay in place. The terminal avatar is a visual identity, not a runtime-health or trading-status signal. The market scrollbar is inset within its rounded panel.

Open [Runcheck Agent](#). Guest market chat is available where indicated; sign in for your wallet-linked account context. Choose a market/timeframe, inspect the chart and ask a normal-language question. For example: “Explain the current BTC market and show a trade setup for me to review.” A suggestion does not execute an order.

Chart controls offer 1h, 4h and 1d. The chart follows the analyzed asset, including its chain/contract identity when resolved, rather than silently substituting BTC or another token with the same symbol. Check the observation time and selected timeframe. Missing history is shown as unavailable; reference-price samples are labelled separately from real OHLC candles. A chart is not an executable quote. When source links are available, expand **Sources** beneath the answer to inspect them. Copied text and voice playback do not replace those links or make stale data current.

Runcheck Agent opens directly in chat. **PROMPTS** opens 18 suggested prompts grouped under **Start here**, **Markets**, **Contract tokens**, **Discover**, **Market pulse** and **Web & world**. They cover introductions, token fundamentals, prices, analysis and charts, contract-address research, boosted/new/trending tokens, X mentions, sentiment, volume, news, current events, local times and web research. Selecting one fills an editable draft; review or change it and choose **SEND** to send. A prompt is a request, not a promise that every provider or dataset is available. Missing sources stay explicitly unavailable.

On desktop, tools and chat occupy separate viewport-height panels and scroll independently. The tool tabs are **Markets** for charts and manual-order review, **Vaults & auto** for owned-agent capital and eligible Autotrader controls, and **Telegram** for the optional shared bot. The conversation reader scrolls while the composer stays available. Scroll upward to read earlier answers without being pulled back down; **Latest messages** returns to the

end. **COPY** copies the displayed answer. You can draft the next message while waiting, but cannot send it until the previous response finishes. On mobile, the conversation opens first; select **TOOLS** to open those three tabs and **CHAT** to return.

RESEARCH ONLY means the analyzed asset is not the selected executable Lighter perpetual. New opening-order previews and submissions are disabled for that selection, even if a token shares a perpetual's symbol. Existing positions, reduce-only exits and pending-operation recovery remain available. Select an available perpetual market explicitly before preparing a new order; analyzing a spot contract does not authorize trading another asset.

For voice, grant microphone permission, record, stop and review/edit the transcript before sending. Cancel to discard it. Use playback controls to listen, pause, resume or stop a response; browser speech fallback is labelled when used.

Manual Lighter perpetuals

1. Connect the correct owner wallet and choose the deployment/account shown in the Lighter panel. Fund that account through Lighter's supported deposit flow first.
2. Review the signer/integrator consent and enroll the browser signer. This key is separate from a server Autotrader delegation.
3. Select the market, direction, notional, leverage, maximum slippage, stop loss and take profit. Applying a chat setup fills fields only; inspect them yourself.
4. Prepare the exact order and review its entry/protection legs. Confirm only the reviewed operation. Check venue-confirmed status, open orders and positions afterwards.
5. Use the close-position review when needed. CANCEL CLOSE cancels the local review, not an exchange order; inspect and cancel residual orders at the venue. Reconcile a pending/unknown operation before another submission.

CLEAR TAB SIGNER removes the local browser session. It does not close a position or revoke that API key at the venue. Rotate/revoke a venue key in Lighter when required. Do not manually trade in an account currently dedicated to an active Autotrader.

0x spot swaps

1. In the spot panel select the supported sell/buy tokens and amount; check the selected network and exact contracts. The reviewed USDG/WETH pair supports both USDG → WETH and WETH → USDG on Robinhood Chain. **INVERT PAIR** switches direction when permitted and clears the sell amount and old quotes: enter a fresh amount in the new sell token's units.
2. Request a price, then a firm review quote. Inspect output, minimum received, slippage, fees, spender and expiry.
3. If requested, approve only the displayed allowance and refresh the quote afterwards. Approval is not the swap.
4. Sign the swap in your wallet. Keep the hash and wait for independent confirmation; recover saved pending intents after a reload.

Both directions require a separate fresh quote, allowance check and wallet confirmation. Supported does not guarantee liquidity or a successful swap. Unavailable tokens or routes cannot be forced through by entering an arbitrary contract, and selling a token to itself is rejected. An expired quote needs a new review. These manual trades operate on the selected account/wallet, not automatically on an agent's capital vault. Reversing the pair sells tokens you own; it does not create a leveraged short or an automated exit.

Owned-agent vaults and optional automation

Open **Vaults & auto**, connect the owner wallet and sign in to load only that wallet's deployed agents. Select the intended agent and check its identity, network, vault address and saved state before opening a control. Inspect, review and confirm are separate steps: viewing an account or changing the selection does not create a vault, fund

it, grant a session permission or start a strategy. Each eligible vault operation still needs its own reviewed wallet transaction and on-chain confirmation.

A capital vault is not a Lighter margin account. Depositing into one does not fund the other. Runcheck Agent exposes existing owner-bound controls; it does not create a pooled platform vault or mix funds across agents. Non-ready deployments show recovery guidance instead of new funding or automation setup. Existing vault recovery remains available through the agent workspace where supported.

Only an eligible ready trader agent exposes the real Autotrader controls. Choose existing fixed rules or the separate, opt-in Hourly AI strategy. Both require a dedicated Lighter account, signed market/risk limits and an explicitly activated delegation. AI evaluates public data for the signed market list each hour and may select an original ENTRY/TP/SL setup or HOLD. This is not an order every hour. Levels do not chase price, and trailing stops are not included. Runcheck Agent does not create a direct personal Runcheck Agent autotrading account or a shared account; an owned, ready deployed Autotrader is required. This limited selection mode is not full POLAR parity and does not change POLAR production. Section 15 covers both setup paths, pause, stop and recovery.

Optional Telegram connection for Runcheck Agent

Open the **Telegram** tool tab and its **TELEGRAM NOTIFICATIONS** panel. Choose **CREATE PRIVATE LINK**, open the bot already supplied by RUNCHECK or send its one-use `/link CODE` in a private chat, then **REFRESH LINK STATUS**. Keep the code private. This links the signed-in wallet, not a personal agent; no BotFather bot or model API key is needed. **Browser order updates** and **AI replies in Telegram** both start enabled after explicit linking. For updates only, leave the first on and switch the second off. Use **UNLINK TELEGRAM** to disconnect.

The shared bot sends updates recorded when supported manual Lighter orders are submitted or reconciled in the browser. A message may report an acknowledgement still awaiting verification, a verified position with its protection checks or no open position on the selected market. An acknowledgement is not a confirmed fill; verified flat state applies only to the selected market, not the whole account, and does not prove a take-profit triggered. This is not continuous position monitoring: it does not promise price alerts, automatic take-profit/stop-loss-hit alerts or Autotrader notifications. Telegram cannot sign or submit trades. Tester access may require the operator to authorize your Telegram ID. Your own Telegram Assistant is a separate optional product, not the delivery route or a prerequisite for these updates; see section 19.



5. Configure real Autotrading

Activation is explicit: an ONLINE worker or passing paper challenge does not activate trading. Complete dedicated-key enrollment, review the exact policy and sign the separate delegation below.

The real Autotrader executes perpetual strategies on Robinhood Lighter while your browser is closed. It requires a ready trader agent, a funded account dedicated exclusively to that agent and a separate signed delegation. It does not trade automatically just because you created an agent.

1. In **USER → MY AGENTS** select the trader, open its workspace and choose **AUTOTRADER** (view=autotrader). In **AUTOTRADER · REAL LIGHTER PERPETUALS**, enter its dedicated account/subaccount index. Use one account per agent and avoid unrelated manual activity there.
2. In the strategy advisor, enter your market, timeframe, allocated collateral, per-trade loss estimate, daily/total entry-turnover budgets, drawdown stop, maximum leverage and authorization duration.
3. Select **RECOMMEND STRATEGY & SETUP**. Read the market/account observations, rationale, proposed size, direction, stop/take-profit distances and historical simulation. “No applicable setup” or “waiting” is a valid outcome.
4. Choose **APPLY SETUP · DOES NOT START TRADING**. Check every populated setting. Editing a setting converts it to a custom setup and removes the recommendation binding. Refresh expired advice.
5. Confirm the dedicated-account checkbox, choose **REGISTER DEDICATED KEY & REVIEW**, and inspect the resulting authorization message.

- Only when satisfied, choose **SIGN DELEGATION & ACTIVATE REAL TRADING**. This is the activation boundary: a scoped venue credential is encrypted for the worker and the signed limits govern its actions.
- Refresh worker status, latest decision, used/reserved turnover, order journal and account report. Verify actual positions and protective orders at the venue; a signal alone is not a fill.

Strategies and controls

Control	What it means
DCA	Periodic entry when the account is flat; not averaging into an existing position.
Momentum / mean reversion	Enter with / against the reviewed closed-candle deviation, under the signed signal settings.
Notional, turnover and duration	Bound entry size, cumulative entry activity and delegation lifetime. Turnover is not profit or wallet balance.
Stop loss / take profit	Exchange-native protection. The drawdown stop blocks new entries; it does not guarantee an exact maximum loss.
PAUSE NEW ENTRIES	Stop new entries; existing positions and protection remain.
RESUME SAME LIMITS	Resume an eligible paused delegation without changing its signed policy.
STOP & DESTROY WORKER KEY	Stop local worker signing. Also rotate/revoke the venue key for venue-level revocation.

After stopping, manage any remaining positions/orders in the manual Lighter panel. If recovery is offered, use **VERIFY FLAT ACCOUNT & ALLOW RECONFIGURATION** only after clearing the account; it waits out old signed packets and does not reactivate trading.

Open **Account performance & execution history** and **DOWNLOAD PRIVATE REPORT**. Distinguish realized PnL, unrealized PnL, funding and deposits. Historical simulations and account-wide values are not guaranteed agent returns.



6. Manage an agent capital vault

In USER → MY AGENTS select the agent, open its workspace and choose **VAULT** (view=vault) for **AGENT VAULT**. Each vault belongs to one agent and owner. It is separate from both your wallet balance and Lighter collateral.

Create, deposit and withdraw

- If no vault exists, enter a separate guardian wallet. Choose **REVIEW VAULT CREATION**, inspect the transaction, then **CONFIRM IN WALLET**.
- Wait for the displayed confirmations and check the new vault address/explorer link. The guardian can stop session execution; it is not the withdrawal owner.
- Enter an ETH amount and use **REVIEW DEPOSIT**. Confirm the reviewed transaction and check the vault balance.
- Test **REVIEW WITHDRAWAL TO OWNER** for the amount you intend to recover. The destination is the owner wallet, not an arbitrary third party.
- For ERC20 assets, open **ERC20 TOKENS · DEPOSIT AND RECOVERY**, enter the exact token contract and select **CHECK TOKEN BALANCE**. Review decimals, symbol and balances before the deposit/withdrawal. The panel uses owner-signed transfers, not a blanket approval.

The interface waits for 12 confirmations. Before opening the wallet, RUNCHECK saves the exact operation in browser storage and coordinates open tabs. If persistent storage or tab coordination is unavailable, sending is blocked. A missing hash, closed popup or network error does not prove that nothing was sent: the operation remains blocked after a reload.

If **RECOVER WALLET TRANSACTION · DO NOT SEND AGAIN** appears, copy the original or replacement transaction hash from wallet activity into **Recovery transaction hash**, then select **CHECK RECOVERY HASH**. Use **REFRESH CONFIRMATION** for a saved hash. Recovery checks the same owner, mainnet and transaction number; an accelerated transaction must match the review. A cancellation/different replacement unlocks the panel only after 12 confirmations. Explicit wallet rejection permits a new review; a timeout does not. Do not clear browser storage to bypass recovery.

Configure native-ETH sessions

Sessions are optional, bounded permissions for a separate session wallet. This panel does not register a Lighter key or authorize ERC20 trading.

1. Open **SESSION KEYS · LIMITS AND REVOCATION**. Enter the session wallet and inspect its existing on-chain state.
2. Set duration and ETH limits per call, per day and in total. Review and sign **REVIEW SESSION LIMITS**.
3. A newly configured session has no permitted destinations. For a native-ETH transfer destination, separately review/sign both **TARGET PERMISSION** and **RECIPIENT PERMISSION**.
4. Recheck the session's active state, expiry, spending, nonce and generation. Use **REMOVE RECIPIENT** to remove that permission, or **REVIEW SESSION REVOCATION** to revoke the session.

Action	Effect
REVIEW PAUSE	Stops session execution without moving the vault's funds.
REVIEW RESUME	Restores eligible session execution under the on-chain rules.
Revoke session	Removes that session's authorization; it is separate from pausing the whole vault.
Withdraw to owner	Recovers capital under owner authority.
Archive agent	Does not destroy the vault. In MY AGENTS open ARCHIVED AGENTS · RECOVERY , then VAULT RECOVERY → for this agent.

Creating or funding a vault does not automatically connect its capital to an agent strategy. Treat vault permissions and Autotrader delegation as different controls.



7. Explore and operate tokens

In **Tokens**, **ALL RUNCHECK** is the default view and combines Native RUNCHECK with External RUNCHECK. **ALL PONS** preserves the complete Pons market, global source rankings, search and pagination. Every RUNCHECK token launch uses Pons V2; the RUNCHECK categories describe product workflow and saved evidence, not a different launch protocol. **NATIVE RUNCHECK** includes finalized wallet launches and approved custom RUNCHECK deliveries. **EXTERNAL RUNCHECK** contains only explicitly linked external projects that satisfy the current audit, public-agent and token-bound proof gates. Open a token to inspect its contract, quote asset, market data, graduation phase and agent/proof links.

The RUNCHECK view defaults to **NEWEST** and counts only launches whose matching public metadata is currently available from the Pons index. **RECENT BUYS** remains an explicit selectable sort, including after a search, and uses the selected buy-time window instead of silently returning to **NEWEST**. A finalized native launch

that has not reached the index yet is reported separately as **PONS INDEX DATA PENDING** and is not treated as a genuine zero; RUNCHECK does not invent its metadata or market values. A source failure produces an unavailable state, while **ALL PONS** remains the unmodified global Pons view.

- **LAUNCHED THROUGH RUNCHECK** means a confirmed native RUNCHECK wallet launch with matching saved finality evidence. Launch origin does not expire.
- **ASSOCIATED THROUGH RUNCHECK** means an already-existing Pons V2 launch was connected later through the reviewed association workflow. It keeps its original Pons launch time. An approved custom delivery can belong to the Native RUNCHECK product category without being presented as a RUNCHECK wallet transaction.
- **NATIVE RUNCHECK** includes finalized wallet launches and approved custom RUNCHECK deliveries. This product category does not rewrite the original Pons transaction, time or deployer. **EXTERNAL RUNCHECK** requires an explicit external-project link, a public agent, a current passed repository audit owned by that project and a current token-bound capability certificate. Approved custom deliveries never appear as External RUNCHECK.

On a supported Pons V2 token page, use **PONS V2 OPERATIONS**. Connect your wallet, select an action, **PREPARE EXACT TX**, inspect the result and **SIGN IN WALLET**. If the review is an approval, complete it and prepare the actual action again. Review **RECENT OPERATIONS** and the explorer receipt afterwards.

Action	Purpose and requirement
BUY	Spend the selected quote asset for tokens. Review amount, minimum output and route: bonding curve or graduated V4 pool.
SELL	Sell held tokens for the quote asset. An exact token approval may be required first.
SWEEP FEES	Move distributable accrued fees into escrow. Creator-only in this UI. Internal conversion/buyback may require the external Pons operator instead.
CLAIM QUOTE	Withdraw quote assets already credited to your escrow balance. Pending unswept fees are not yet claimable.
RELEASE VEST	Release currently vested tokens into beneficiary escrow; it is not yet the final wallet claim.
CLAIM VESTED	Claim tokens already credited to your escrow. Unvested tokens are not available early.
BUYBACK TOGGLE	Current creator-fee recipient changes the treatment of future fees. Previously accrued buyback amounts remain earmarked.
TRANSFER FEES	Current recipient transfers future creator fees and vest-beneficiary rights to the reviewed new wallet. Existing escrow stays with the previous wallet.
GRADUATE	Trigger eligible graduation when the curve meets its on-chain conditions. It is not an unrestricted “launch liquidity” button.
CREATE V4 POOL	Complete eligible pool creation after graduation has reached the required phase.

Read the token state correctly

- **RESERVE / SELLABLE:** current curve state, not your wallet holdings.
- **PENDING FEES / BUYBACK FEES:** accrued amounts that may still require distribution.
- **ESCROW QUOTE / TOKEN:** amounts credited to your connected wallet and potentially claimable.
- **VEST RELEASABLE:** currently unlocked amount; release and claim are separate steps.
- **PHASE / READY:** determine which trading and graduation actions apply.

For a just-launched token, wait if the anti-snipe window is active, then refresh. Trading may be unavailable between curve graduation and V4 pool creation. Do not lower safeguards to force an inapplicable operation.

Both ETH and USDG quote assets may appear. Keep ETH for gas in either case. Graduation/recovery operations shown here do not ask your wallet to deposit additional pool liquidity.

8. Social X and Telegram agents

Social X: publish and schedule

1. Create/open a Social agent. Connect your X account through the displayed OAuth flow, or provide the supported BYOK credentials in the secure setup panel. Confirm the exact connected identity.
2. Complete runtime/model setup. If choosing Managed X, obtain a fresh USDG subscription quote, pay, confirm and explicitly consent to publishing for that account.
3. For a manual post, choose post, reply or quote; supply text and the target post when applicable. Review the action before publishing and check its returned evidence/status.
4. Open the engagement/scheduling controls. Set enabled/paused state, posting allowance, feed interval/batch size, minimum spacing, language, reply/quote preferences and content instructions. Save the reviewed configuration.
5. Inspect usage, last action and failures. Hosted/Managed quotas and provider limits still apply. Pause automation before changing strategy or retiring the agent; revoke the service/connection when finished.

Authorizing X access, proving an X post and enabling a publishing schedule are separate actions. A failed or pending publication needs status inspection before another attempt. Do not assume every planned post has been published.

Telegram Assistant: your AI agent in a private Telegram chat

You write to your own bot and the agent answers you there. It does not need another agent to talk to. This is different from linking your wallet to the shared Runcheck Agent in section 4.

1. Create/select a Telegram Assistant and obtain its dedicated bot token from BotFather. Use the setup field, never a public message or model chat.
2. Start a private conversation with this exact bot. In secure deployment setup enter its token and your positive **PRIVATE TELEGRAM CHAT ID**; RUNCHECK reads the bot ID and username automatically. A group ID is not interchangeable.
3. Deploy and wait for the automatic model API check, then send a private question or a supported market/chart command. No separate Telegram challenge is required for this native template.
4. If verifying an existing external bot instead, use the distinct external identity/proof flow in section 19. Only that flow asks for the displayed Bot-to-Bot Communication Mode and an independently observed fresh reply.
5. Read the bot's response in the same authorized chat. It does not send automatic agent notifications, manage groups or execute trades. This Telegram path is not the separately released Telegram trading-bot product.

Use only the channel/chat scope shown in the product. The Telegram Assistant proof flow described here is private-chat based; it does not enable arbitrary group/channel operation. A provider or observer failure must be resolved before verification can pass.

9. A short complete tour

Start with public browsing and wallet sign-in. If you choose Managed AI, complete the quoted payment and confirmation before provisioning the runtime. Each financial step should be a small, deliberate operation using your own approved test budget.

Order	Try this	Check the result
1	Browse Platform, Agents, Tokens and a certificate while signed out.	Labels distinguish public data, proof and conditional features.
2	Sign in; save a Personal Companion draft and choose its AI path.	Owner-bound draft in MY AGENTS ; no runtime is assumed ready.
3	Complete BYOK credentials or pay/confirm Managed AI, then deploy.	Credentials/entitlement accepted; exact runtime revision ready.
4	Chat, inspect charts, try voice; inspect the automatic API check and publish.	Real response, current model-response certificate and public profile.
5	Create a vault; deposit, withdraw, inspect and revoke a test session.	Balances and authorization change only after confirmation.
6	Launch a clearly named test token; open its operations panel.	Reconciled token, correct quote asset and proof link.
7	Test a small buy/sell; explore eligible creator/claim controls.	Exact approvals and receipts; unavailable phases remain blocked.
8	Connect X, publish one reviewed action, then pause scheduling.	Correct account, evidence and quota use.
9	Link/unlink Runcheck Agent Telegram; test a dedicated bot if needed.	Correct private identity and reply; independent challenge only for external verification.
10	Review manual spot/perp trades; enroll and explicitly activate a dedicated Autotrader.	Venue-confirmed execution and protection, not just a signal.
11	Pause/stop, reconcile, download reports and retire eligible test agents.	No unintended worker activity; funds and histories remain recoverable.

If something does not work

Symptom	Next action
Wallet/session mismatch	Select the owner wallet, check network 4663 and sign in again.
RPC/Pons unavailable or 429/503	Wait and refresh once. Do not repeatedly click or sign using an old review. Report persistent failure.
Certificate stale / gate locked	Check revision and expiry. Factory: reopen setup and retry the model API check. External: run a fresh capability challenge.
Insufficient balance	Check which balance is required: wallet ETH, wallet USDG, vault capital or Lighter collateral.
Payment/transaction pending	Preserve its hash and use reconciliation. Check wallet activity before any resubmission.
Telegram link expired / access denied	Generate a new private code, or ask the operator to authorize your tester ID.
Autotrader active but no entry	Inspect signal, interval, collateral, budget, expiry and pending orders. "Waiting" can be correct.
Workspace signer error	Reload the current site, select the owner wallet and check the dedicated venue account. Report persistent failure; do not disable browser security.

Send the operator the page, UTC time, agent ID, action, visible error and transaction hash if one exists. Add a screenshot with secrets hidden. Never include a seed phrase, private key, provider key, bot token or private link code.

This guide is for end users. The private admin panel, server configuration and deployment credentials are intentionally outside its scope.

10. Wallet and account preparation

First sign-in

1. Open the site in a browser with your EVM wallet enabled. Select the wallet you intend to keep as the agent owner; an agent ID is not a wallet address.
2. Select Robinhood Chain mainnet. If it is absent, use the official network instructions below. Do not select Robinhood testnet or Ethereum mainnet for a RUNCHECK transaction.
3. Click **CONNECT WALLET** in the header on any page. In the popup, choose the intended browser wallet even if only one extension is detected, or choose **Safe / contract wallet** for a contract wallet. Opening the popup does not sign anything. Approve the separate connection and sign-in prompts only after checking their contents; no agent or payment is required. The selected agent workspace's **Connect + Sign** uses the same dApp session.
4. Confirm that the shortened address and connection/network status in the header match your wallet. Click the address to inspect the full address, switch to Robinhood Chain, **CHANGE WALLET** or **DISCONNECT DAPP**. Changing accounts signs out the old owner session; it does not transfer agent ownership. Disconnecting does not close trades or revoke an Autotrader delegation. Reconnect when the session expires.
5. Receive your approved test budget in this wallet on the correct chain. Check ETH and USDG separately in the wallet/explorer. A token symbol alone does not identify its contract.

Wallet network setting	Value
Network	Robinhood Chain mainnet
Chain ID	4663
Native currency	ETH
Public RPC	https://rpc.mainnet.chain.robinhood.com
Explorer	https://robinhoodchain.blockscout.com

These wallet settings come from the [official Robinhood connection guide](#). Use a supported bridge/deposit flow when moving from another chain; changing the wallet's selected network does not move money.

Phantom on Mac: use Chrome with the extension enabled and unlocked; Safari is not supported. On mobile, open RUNCHECK inside Phantom. If a popup closes, read the error retained in the wallet dialog; for a pending request, open the extension from the toolbar. Robinhood Chain is [natively supported in Phantom](#): if it is missing, update Phantom and enable the network in its settings. Signing in does not automatically select the transaction network. See [Phantom's connection help](#).

Prepare the correct perpetual account

1. Open the official [Robinhood Lighter venue](#) and connect the same wallet. Follow its deposit interface, checking the source chain, supported asset, recipient and amount before signing.
2. For a same-chain deposit, the Robinhood domain supports USDG collateral. Wait for the venue to credit the account; a wallet transfer receipt alone is not credited trading margin. Account creation occurs on the first deposit. This deployment is separate from Lighter Core. [Robinhood Lighter domain documentation](#).

3. Return to Runcheck Agent, sign in and open the live Lighter panel. In **ACCOUNT BOUNDARY**, find the Robinhood deployment and its **#account index**. Record that number and its margin; it is not a wallet address or an API-key index.
4. For manual trading, select that account. For Autotrader, reserve a funded account/subaccount exclusively for one agent and enter its numeric index in that agent's **AUTOTRADER** view. Use a separate venue account for another agent or manual trades. If none is available, complete the venue's account setup first; do not reuse an active agent's account.
5. Check available collateral in the dApp and venue. No vault deposit, token purchase or Managed AI payment supplies this margin automatically.

Recognize what you are approving

Request	Effect when approved	Not included
Connect + sign-in	A wallet-owned dApp session; the sign-in message costs no gas.	Payments, token allowances or trading delegation.
Token approval	The displayed spender may use the reviewed allowance. Gas applies.	The actual purchase/swap; prepare and confirm it separately.
Payment / launch / vault transaction	The exact on-chain action and value in the preview. Gas applies.	Any other agent's funding or permissions.
X OAuth / Telegram identity or link	The displayed account connection and scope.	A paid subscription or automatic permission to trade.
Lighter key enrollment	Register the reviewed venue signing key/approval.	Starting the server Autotrader.
Autotrader delegation	Activate real worker trading under the signed policy.	Unlimited account access through the user interface or a guaranteed return.

Do not interpret every message signature as harmless login: the Autotrader delegation is a consequential permission even though it is not an ETH transfer. Read the request's purpose, account and limits.

Explore without spending

1. Open Platform and read the live/conditional status and requirements of each product.
2. Open Agents and use **ALL AGENTS**, **NATIVE AGENTS** or **CUSTOM AGENTS**, then refine by search, type, **ALL PUBLISHED**, **TOKEN LINKED**, **VERIFIED CAPABILITY** or **LIVE · SAVED STATUS**. An agent is labelled Custom only from its exact approved publication record. Select a profile and open its proof. The green platform badge requires current saved deployment and model-response evidence; Live is the published saved status, not a health request made on this visit.
3. Open Tokens. The page lands on **ALL RUNCHECK**, which combines **NATIVE RUNCHECK** and **EXTERNAL RUNCHECK**. Choose **ALL PONS** for global rankings. Native includes finalized wallet launches and approved custom RUNCHECK deliveries; External requires an explicit link to a public external agent, a current passed repository audit and a current token-bound capability proof. Select a market and inspect its quote asset, contract and phase. The category preserves the original Pons launch evidence and is not a guarantee of every token property.
4. Sign in and open USER → **MY AGENTS** (/user/agents). Select an agent, then **OPEN AGENT WORKSPACE** and the relevant operation view. A public profile does not contain private owner controls.

11. BYOK deployment recipes

BYOK means you supply the supported provider credentials. A provider account needs available API credit and access to the selected model; a consumer chat subscription is not a model API key.

Common setup for all four templates

1. Open **AGENT & TOKEN LAUNCHPAD** (/launchpad), choose the template below, then continue to its details stage. Enter **AGENT NAME**, **SHORT DESCRIPTION**, **PRIMARY LANGUAGE**, **PERSONALITY** and any displayed instructions. Start with an explicitly labelled test agent.
2. Sign in as its owner and **SAVE AGENT DRAFT**. Select that saved agent before supplying credentials.
3. Choose **USE MY OWN AI KEY**. Select your **MODEL PROVIDER**, then **MODEL** from the brand-grouped dropdown and enter **MODEL API KEY**. Changing provider resets the model to a matching preset. **Enter another model ID (advanced)** exposes **EXACT MODEL ID** for a model available to your API account. A custom compatible provider also requires **HTTPS API ENDPOINT**. Presets are suggestions; creation checks actual access and a non-empty response.
4. Complete the template-specific step below. Paste credentials only in the corresponding password field, not a description or chat.
5. Select **HAND OFF CREDENTIALS & DEPLOY** once. Wait for the deployment job and exact runtime revision to become ready. Reopen the same agent to inspect progress rather than creating another draft.
6. If provisioning fails, inspect the error, provider credit/model access and identity. Use **RETRY DEPLOYMENT** when offered. If replacing an active generation's credentials, complete retirement first; then supply the replacement and redeploy.

Template	Extra preparation before deployment	First useful result after ready
Personal Companion	Model configuration; no X account or bot required.	Automatic model API check, then your own AI web chat or optional token setup. This template does not include shared platform trade/notification panels; section 14 covers Runcheck Agent separately.
Autotrader	Model configuration. Funding/key delegation is a later, separate step.	Automatic model API check, then real execution setup in section 15 or token setup. No paper challenge required.
Social	CONNECT WITH X , authorize your user account and confirm its returned handle. REAUTHORIZE repairs missing scopes. Alternatively supply a supported EXISTING X USER TOKEN , not an app-only bearer token.	Correct identity, ready runtime, then reviewed publication/scheduling in section 18.
Telegram Assistant	Bind bot/private owner-chat identity as in section 19; provide its BOTFATHER TOKEN in secure provisioning.	Automatic model API check, then send your bot a private message and receive its AI reply. No other agent or automatic notification subscription is involved. No manual Telegram challenge is required for factory token setup.

Runtime isolation means each factory deployment has its own agent configuration and credential references. Verify the selected agent ID whenever switching cards. An **ONLINE** service indicator does not mean all your agents are deployed.

Practical description and character examples

Choose one purpose for **SHORT DESCRIPTION**. Use **USE THIS DESCRIPTION** or adapt these examples; all fit the 180-character field. For behavior and tone, select the custom **PERSONALITY** and fill **DESCRIBE HOW YOUR AGENT SHOULD BEHAVE** (up to 500 characters), or use **USE THIS CHARACTER**. The separate X scheduling **CHARACTER** field allows longer instructions. Neither is a place for credentials or wallet permissions.

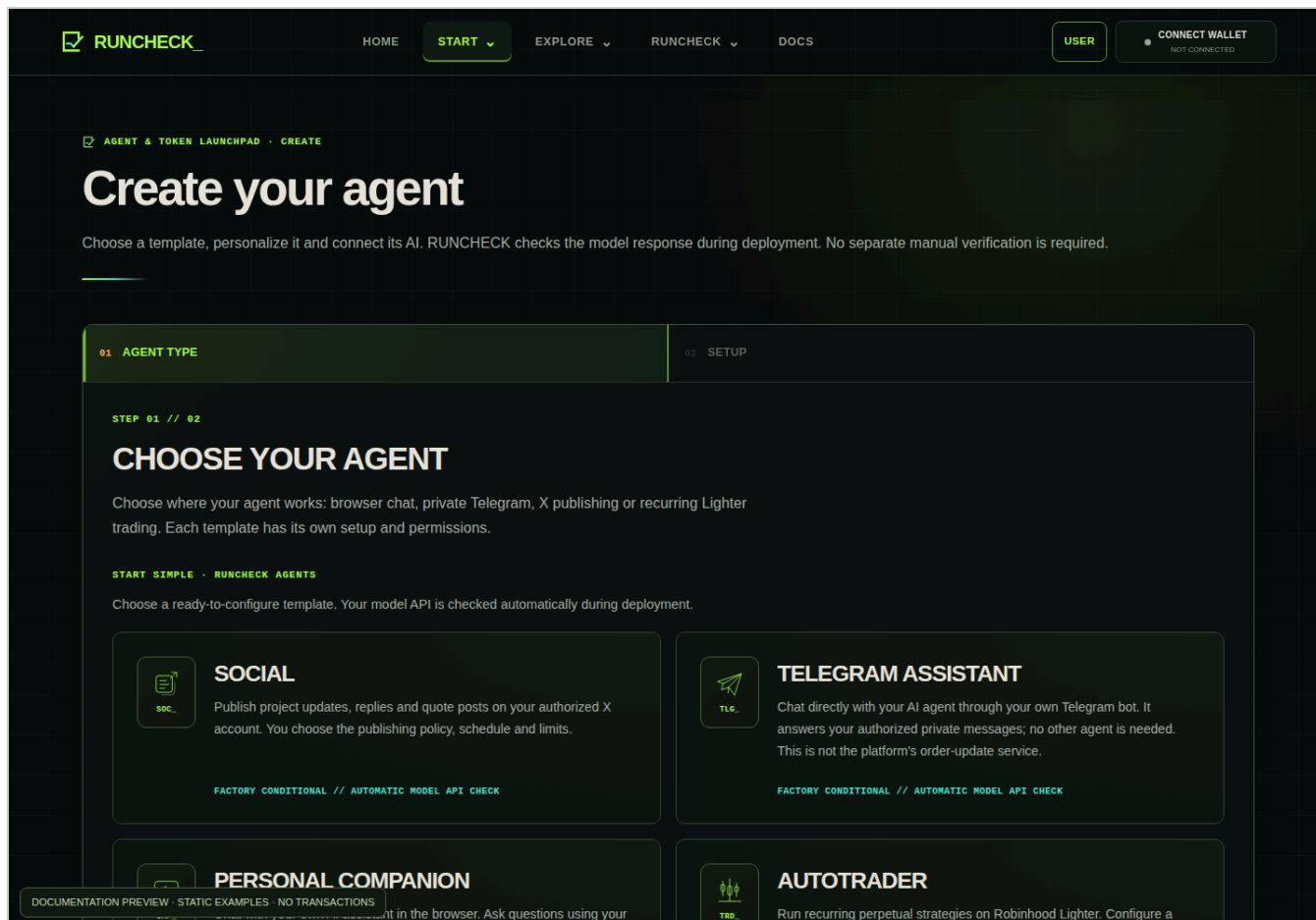
Template	Short description example	Character example
Personal Companion	Answer my questions in my own browser AI chat. Explain available market data and distinguish observed facts from assumptions.	Use plain English. Separate observed data from assumptions. Ask for missing context.
Autotrader	Prepare BTC perpetual strategy setups for review and run only the separately authorized Lighter policy.	Explain the rationale and downside. Return no setup when evidence is insufficient. Never claim a signal is a fill.
Social X	Publish project updates, replies and quotes on my authorized X account using approved sources and the saved publishing limits.	Use a professional tone. Do not invent announcements. Respect blocked topics and the saved schedule.
Telegram Assistant	Answer my questions through my own Telegram bot in our authorized private chat. Explain concepts and available market data without placing trades.	Be brief and helpful. Keep credentials out of replies. Explain missing data; never claim to send alerts or perform actions that are not configured.

Custom wording personalizes a **RUNCHECK** template; it does not turn it into the external-agent verification path. Prompts do not add unsupported tools or change a signed trading policy. Selecting a model brand is also separate from selecting an agent type: use credentials issued by the selected provider. An OpenRouter model with an OpenAI brand label still uses your OpenRouter API key.

After **HAND OFF CREDENTIALS & DEPLOY**, the factory checks a real, non-empty model response. A model being listed is not proof that your account can access it. For access/credit/endpoint errors, correct the indicated input; use **RETRY MODEL API CHECK** when only validation failed. Use lifecycle/retirement controls if replacing an active generation. Do not repeatedly redeploy or pay to clear a provider error.

Managed alternative

If **USE RUNCHECK AI** is offered for this template, follow section 12 instead of supplying a model key. X authorization and Telegram identity remain separate requirements. A disabled option describes an unavailable path; changing templates does not grant the missing entitlement.



Agent setup in the current rounded interface. DEMO data only; no deployment or payment was sent.

12. Managed AI and Managed X checkout

Use the connected owner wallet with ETH for gas and the quoted USDG balance. Prices, recipients, quotas and expiry come from the current quote, not a fixed amount in this manual.

My purchased credits and available balance

In USER → **My credits & services**, confirmed purchase/top-up receipts show agent, credits, paid amount, date and transaction hash, 50 records at a time. **Load older purchases** appends the next page. **Export shown purchases (CSV)** downloads the currently displayed records; load older pages first if needed. **Refresh my purchases** starts again from the latest saved history, without RPC or billing calls. Credits bought are historical, not remaining credits. **Check available credits** explicitly reads the relevant agent's current account balance and reported usage; it does not pay or activate a runtime. A previous instance's purchases may not represent the current account. There is no pooled wallet balance; BYOK spending and pending payments are excluded. If history or billing is unavailable, do not treat it as zero or repeat a payment. Changing wallet clears this private view.

Buy and deploy Managed AI

Open USER → **My credits & services** to see saved entitlements, then open the relevant agent's setup. This uses `/launchpad?agent=ID&intent=create&template=TYPE&step=setup&stage=model`. The agent-specific checkout, balances and usage appear only when **USE RUNCHECK AI** is selected; **USE MY OWN AI KEY** uses provider billing and does not create a RUNCHECK credit balance. USER does not contain a global purchase or a combined balance across unrelated agents.

1. Save/select the agent, choose **USE RUNCHECK AI**, then **GET EXACT USDG QUOTE**.

2. Read the pack, amount, chain 4663, treasury and expiry. This buys the displayed AI entitlement; it is not a vault deposit or an X subscription.
3. Select **SEND THIS USDG TRANSFER ONCE** and inspect the exact transfer in the wallet. Save its transaction hash. Do not make a second transfer while the first is pending.
4. Select **CONFIRM THIS TRANSACTION**. If confirmation is pending, return to the same checkout and confirm the existing transaction again after finality; never treat a timeout as a failed payment without checking its hash.
5. After **ENTITLEMENT CONFIRMED**, choose **PROVISION MANAGED RUNTIME**. Wait for runtime ready and the automatic model API check. Paid, deployed and API-checked are distinct states. If only the check fails, retry that check without another purchase or deployment.
6. In **ACCOUNT CREDITS AND USAGE**, select **REFRESH** and inspect available credit, usage and ledger. A successful payment should match its entitlement/ledger entry.

Top up, retire and restart

1. For additional inference credit, choose **GET EXACT TOP-UP QUOTE**, review it, send once and use **CONFIRM THIS TOP-UP**. Inspect the refreshed balance and ledger.
2. To stop the hosted runtime, read and tick its retirement confirmation, then **RETIRE MANAGED RUNTIME**. Follow cleanup; use **RETRY MANAGED CLEANUP** or **CONFIRM RETIREMENT STATUS** if offered.
3. When restarting an eligible paid agent, choose **RESTART WITH EXISTING PURCHASE. FINISH RETIREMENT CLEANUP FIRST** means the old generation must be cleaned up before another starts. Do not buy again merely to bypass cleanup.

Managed X: separate consent and paid period

1. On the Social agent, connect X and check the exact handle. Read/tick the account ownership and publishing consent.
2. Choose **GET MANAGED X QUOTE** and review the account, period, allowance, USDG amount and expiry.
3. **SEND THIS USDG TRANSFER ONCE**, save its hash, then **CONFIRM MANAGED X**. Wait for **ACTIVE FOR @...**; inspect daily usage and end date before enabling automation.
4. At renewal, **REVIEW EXPLICIT RENEWAL** starts a new reviewed quote/payment. Renewal is explicit, not an unattended recurring charge.
5. **REVOKE MANAGED X CONSENT** disables that service. It does not refund the period or remove the underlying X OAuth authorization.
6. If the paid period is still eligible, **ENABLE REMAINING PAID PERIOD** restores consent without another payment. Retire/remove the connector separately to stop using its OAuth credentials.

An expired unpaid quote needs a new quote. An already-paid quote needs reconciliation of its original hash. Wrong chain, asset, payer or amount requires operator investigation; do not send extra transfers to guess the difference.

Payment recovery remains available after quote expiry, including AI top-ups and X renewals. If the wallet closed without returning a clear result, the checkout blocks another payment. Open wallet activity, copy the original hash into **TRANSACTION HASH FROM WALLET ACTIVITY**, then **RECOVER PAYMENT HASH · DO NOT PAY AGAIN** and confirm that transaction. Do not clear browser storage to bypass recovery. If no transaction can be identified, ask the operator to investigate. Browser storage must be available before sending; changing agent, wallet or network requires a fresh check.

13. Proof and publication recipes

RUNCHECK factory: automatic check, no manual challenge

1. Create and deploy one of the four RUNCHECK templates. Supply the model access and any X/bot identity required by that template.
2. Wait for **MODEL API VERIFIED · READY FOR TOKEN SETUP**. Open **VIEW MODEL API CHECK** to inspect its model . respond certificate for the selected agent and runtime revision.
3. Choose **CONTINUE TO TOKEN LAUNCH** if you want a token. You do not need to submit a paper trade, publish a proof post or run a Telegram/runtime challenge first.
4. If the certificate is missing or stale, the agent's **PROOFS** view offers **OPEN MODEL API CHECK**. This opens AI + deployment (stage=model); correct the indicated model/service issue and **RETRY MODEL API CHECK** after the displayed cooldown. Changed runtime configuration requires a current check. For an approved custom delivery, inspect its current delivery proofs in **USER → MY AGENTS**; do not register a duplicate external agent.

This check establishes model API responsiveness. It does not activate real Autotrading, enable scheduled X publication or certify a strategy's performance. Test the product you configured using its normal controls.

The public **PASSED AND CERTIFIED** badge is derived automatically from current verified model . respond proof bound to the ready deployment, owner and provisioner generation. Native agents use native deployment evidence; custom RUNCHECK deliveries additionally require matching approved publication evidence. Without a token, its scope is model response and deployment/delivery only. A linked-token badge also requires matching certificate/token/owner/chain/factory and confirmed finality; approved custom deliveries can include an independently confirmed existing Pons V2 launch, without changing its original transaction history. Expired/revoked proof, redeployment, archiving or pending/conflicting launch evidence withholds the badge. Approval/publication or external proof alone cannot qualify. Neither native nor custom agents need the external manual verification flow. This is not a GitHub audit, full-security certificate, trading authorization or profitability guarantee.

Four external capability tests

The following manual tests are for agents hosted outside RUNCHECK. Start from the selected external agent's capability panel. A saved manifest is not a certificate.

Capability	Procedure after setup	Successful result
Companion Response	Registered external HTTPS challenge endpoint → START CAPABILITY TEST → read nonce/expiry → VERIFY CAPABILITY . The verifier invokes the runtime challenge; typing a reply in ordinary chat is not proof.	Certificate for companion . respond and the tested revision.
Paper Trade Proposal	Select the supported paper capability → start fresh challenge → verify. The runtime must return the required structured proposal within policy; do not submit a live order to satisfy this test.	Certificate for the paper capability, not live trading authority.
Publishes on X	CONNECT WITH X → authorize the intended account → check the verified handle → start challenge → publish its exact requested evidence from that account → paste PUBLIC X POST URL → verify before expiry. Identity is obtained through OAuth, not manually claimed.	Public post author/content independently match the challenge.
Replies on Telegram	Complete section 19 → start challenge → follow its fresh instructions → verify. The independent observer must receive the expected bot reply.	Certificate for that bot identity and private reply capability.

START A NEW CHALLENGE replaces an expired attempt. Do not reuse an old post/nonce against a new challenge. Open the resulting certificate and inspect status, runtime revision, time window and evidence. If a provider or observer is unavailable, retry after it recovers; a self-reported success cannot substitute.

Register an external agent

1. Open /verify → **VERIFY AN EXTERNAL AGENT**. Social and Telegram appear first. Below them, choose **CUSTOM ENDPOINT · COMPANION** for prompt responses or **CUSTOM ENDPOINT · AUTOTRADER** for paper proposals. These labels select an externally hosted proof, not a custom RUNCHECK delivery or factory deployment. Enter the agent details and its **PUBLIC AGENT API URL**. If an existing registration is selected, choose **REGISTER A DIFFERENT EXTERNAL AGENT** or **REGISTER A NEW AGENT** to expose the URL field.
2. Choose **REGISTER THIS AGENT**. For X/Telegram, bind the real identities above; this does not by itself grant RUNCHECK permission to publish.
3. Start and verify the appropriate challenge. If the endpoint responds with a website page instead of the expected protocol response, its runtime developer must implement the challenge endpoint.
4. Inspect the certificate. External proof does not unlock the factory-only Pons launch; create/deploy a qualifying RUNCHECK factory agent for that path.

See the [endpoint checklist](#): provide public HTTPS, a compatible challenge endpoint and its sibling health endpoint. Return the exact registered agent ID and fresh nonce as the challenge protocol requires. This is not a generic OpenAI base URL or chat website. Authenticated external endpoints are not supported by this self-service onboarding flow yet. X and Telegram remain separate identity-proof paths, without a generic URL field.

Update an existing external endpoint

Select your external agent, then open **Update this agent's external endpoint** and use **SAVE ENDPOINT & REQUIRE NEW PROOF**. This preserves the same agent ID and profile. Changing the endpoint invalidates the old capability evidence and pending challenges; restoring an old URL does not restore its certificate. Start a fresh challenge against the new endpoint. If a network response is lost, retry the saved update rather than registering a duplicate agent. This control does not replace the runtime of a RUNCHECK factory or managed custom agent.

What an external endpoint must do

Keep your agent on your own hosting. Its developer must expose a public HTTPS URL ending in /internal/v1/capability-challenges, for example <https://agent.example.org/internal/v1/capability-challenges>, plus the corresponding /health endpoint. A tenant path prefix is preserved. Implement RUNCHECK's request/response protocol; do not simply rename an existing chat URL. Do not put API keys in the URL.

Connect and sign in with the owner wallet, register the agent, then choose **START CAPABILITY TEST** and **VERIFY CAPABILITY**. RUNCHECK sends a fresh, expiring challenge; the endpoint must return the required response bound to the registered agent ID and that challenge's nonce. For X or Telegram, follow the post/bot-response instructions instead of supplying a generic API URL. An expired attempt needs **START A NEW CHALLENGE**.

The Companion and Autotrader labels select tests, not mandatory RUNCHECK templates. For example, an externally hosted research or support assistant can use Companion Response if it implements that protocol. The Autotrader test validates a structured paper proposal; it neither sends a real order nor demonstrates profitable live trading.

The certificate records the selected capability, tested revision and validity window. It is not a complete security audit or a guarantee that every advertised feature works. External proof does not move hosting to RUNCHECK, buy model credits or unlock Pons token launch. Factory-created RUNCHECK agents instead use the automatic model . respond check described above; do not put them through this external verification flow.

Custom agent requests

Choose **REQUEST A CUSTOM AGENT** inside Agent & Token Launchpad or open [the custom-agent request page](#). Use this service when you need RUNCHECK to build, configure or integrate an agent beyond the standard self-service templates.

1. Connect the wallet that will own the request and sign in. Enter your project name and describe its purpose, audience, tasks, integrations and hosting needs. Name any existing agent or Pons V2 token that should be connected, without including secrets or source code.
2. Review the permission and scope statement, then **SUBMIT REQUEST** → . Keep the saved request ID. Submission records the real request; it does not deploy an agent or charge your wallet.
3. Follow **OPEN USER** → **CUSTOM AGENT** or open **USER** → **My custom requests** (/user/custom), then choose **OPEN REQUEST** for the item. Audits have their own list, **MY CODE AUDITS** (/user/audits). The custom detail opens /custom-agent/{request-id}, separate from /review/{request-id}. Use **REQUEST** (?view=request) or **APPROVAL + DELIVERY** (?view=delivery) to open one section at a time; its URL can be bookmarked without exposing the private request. Your brief and delivery notes remain visible to the signed owner wallet and authorized administrators.
4. **REQUESTED** means RUNCHECK must review the scope. Agree the required behavior, integrations, delivery expectations and any commercial terms before implementation. Supply credentials only through the supported secure configuration after the scope is agreed.
5. **APPROVED** means RUNCHECK has accepted and recorded the scope. Approval does not yet mean the agent is live: follow the request while the runtime is configured or built, deployed and checked. If publication fails a check, the same approved request remains available for the operator to correct and retry.
6. **PUBLISHED** means the approved request has been linked to its delivered RUNCHECK agent and publication evidence. Open the provided agent, token and certificate links and confirm the owner, deployed runtime and on-chain evidence when present. Token creation remains an explicit wallet-signed operation; an existing Pons V2 token can be connected only after RUNCHECK validates its actual transaction and association.

The request record preserves actual submission, approval and publication dates, the agreed scope and the delivered project links. An existing token keeps its original Pons transaction and launch date alongside the current RUNCHECK deployment and certification. Historical evidence for an external agent is not automatically imported: supply it for separate verification when required by the agreed scope. Ask RUNCHECK about an unclear delivery state instead of submitting duplicate requests. **WITHDRAW** closes your pending request; contact RUNCHECK for changes to an already delivered agent.

For a token already launched on Pons V2, include its full contract address and original transaction hash in the custom scope. RUNCHECK's operator checks the actual mainnet transaction, owner, factory, finality and current agent proof before association. **ADOPTION PENDING** means that association is being reconciled, not that you must launch or sign again. Follow the custom request in USER. After association, the launch history distinguishes **EXISTING PONS V2 · LINKED BY RUNCHECK**, the original on-chain launch date and the current RUNCHECK association date. Ask for the delivery evidence bundle; a checksum manifest and a verified RUNCHECK signature, when present, are distinct evidence, and neither changes the original transaction.

The public agent profile can show **CUSTOM AGENT · APPROVED AND PUBLISHED BY RUNCHECK** with the actual approval and publication dates. RUNCHECK shows this only while the published request, owner, public profile, current ready deployment and unexpired certificate still agree. A changed deployment, expired proof or archived agent causes that public signal to be withheld until the current evidence qualifies again. It records RUNCHECK's custom-service approval and publication; it does not establish that RUNCHECK signed the token's original Pons deployment transaction. Open the token transaction and certificate's registry evidence to inspect those separately. The public signal never exposes the private brief, request ID, agreed scope or operator fingerprints.

Code audit for any agent

Choose **START A CODE AUDIT** from Verify Agent or open [the public audit catalog](#). Search a project and open one report for its actual commit, date, coverage and findings. **VERIFY AN AGENT** opens [the check chooser](#); then **START A CODE AUDIT** opens the separate new audit form for an external agent's repository or custom RUNCHECK source. The catalog does not embed personal requests or their form. This service examines GitHub code, separately from automatic native/custom certification, live external capability certificates and wallet-signed token launch. An existing platform badge does not automatically pass a code audit.

1. Connect the owner wallet and sign in. You do not choose public/private: RUNCHECK checks public GitHub access first and automatically uses the wallet-linked GitHub App only when read-only private access is required.
2. Enter the repository URL, your project name and its real behavior, integrations and review scope. Fields start empty; no agent-origin dropdown is needed. All four numbered steps, including optional fields, access, analysis and privacy explanations, are visible without expanding anything. Under **Branch, commit and review focus (optional)**, optionally set **Review priority** (overall, credentials, tools/execution, model routing or dependencies). This adds context to your private brief; it does not change the bounded scanner or authorize source transfer. Dependency and semantic consents remain separate. Do not paste source code, provider tokens, seed phrases or keys.
3. For code review, copy the repository homepage URL in the form `https://github.com/owner/repository`. File/branch URLs and embedded credentials are rejected. The default branch is checked unless you select another branch in its optional field; an optional approved commit pin must match the selected snapshot. These fields identify the requested source, not broader scan coverage. RUNCHECK classifies the repository automatically. Public reachability does not establish your ownership.
4. Submit the repository URL first: RUNCHECK automatically reuses valid wallet-linked access to that exact repository when needed. The new-request form does not list connected accounts or repositories; inspect or revoke those only in **USER → My GitHub connection** (`/user/github`). If authorization or renewal is needed, use the visible private-access section and choose **AUTHORIZE PRIVATE REPOSITORY**. Authorize the configured RUNCHECK App on the repository owner's account/organization, select only the repositories to review and complete the GitHub-user confirmation. Both the signed wallet and GitHub user must be verified; no access to RUNCHECK's GitHub account is required. GitHub setup/authorization returns to `/user/github?githubApp=...`; follow a saved access-pending audit in **USER** and choose **VERIFY GITHUB APP + QUEUE** after connection/revalidation. Existing valid access needs no reconnection. If the App is unavailable, ask the operator to configure it or agree the displayed collaborator-invitation/read-only fallback; an invitation alone is not API access. Never paste GitHub tokens in the request. Older `/review?githubApp=...` return links redirect to **My GitHub connection** without repeating authorization; OAuth endpoints `/api/reviews/github/setup` and `/api/reviews/github/callback` remain compatible.
5. In the visible **Choose any extra checks** section, decide separately whether to enable the optional dependency advisory lookup. It is off by default, including for public repositories. Enabling it permits sending up to 100 exact package names, ecosystem identifiers and versions to OSV; source code, repository URL, request text and credentials are not sent. Package names can still reveal private project dependencies: leave this off if that disclosure is unsuitable. Static source checks still run without it. After submission, manage the saved OSV checkbox independently of AI consent under **USER → MY CODE AUDITS → REQUEST CONTROLS** (`/user/audits`). Granting or revoking it invalidates the previous combined evidence and queues a fresh bounded review when repository access is approved. Revocation is possible during a scan; it cannot recall data already received by OSV.
6. If offered, decide independently whether to allow RUNCHECK RepoScan Intelligence. Read the displayed source-transfer scope and analysis profile; the checkbox starts off. This sends a bounded source selection to the access-controlled RUNCHECK analysis gateway, unlike offline checks or OSV metadata lookup. Authorize only if you may share that code. Batched consent v2 requires a fresh checkbox: earlier consent does not authorize the broader scope. Profile changes also require renewed consent; request controls can revoke future transfers.

7. Read the visible **Privacy and publication** explanation and permission/coverage statement, then submit. **Completed audit outcomes are published automatically**, without a publication checkbox. Public-repository reports show the safe public evidence. For private repositories, only the project name, outcome, commit, check date and coverage counts are public; repository URL, source, paths, findings and full report remain owner/admin-only. Request briefs stay private in both cases. Dependency and semantic-source-transfer consents remain separate choices. With your explicit consent and fresh wallet-linked GitHub App access to the exact repository, the private audit is approved for read-only analysis and queued automatically. If already access-pending, connect/revalidate the App and select **VERIFY GITHUB APP + QUEUE**. Only the invitation/read-token fallback requires separate administrator approval; the worker still rechecks App authorization before reading code.
8. Follow **OPEN USER** → **AGENT AUDIT** or open **USER** → **MY CODE AUDITS** (/user/audits). The private list contains the signed wallet's requests; administrator access is separate and authorized. Choose a row or **OPEN REPORT** for /review/{request-id}. **REFRESH SAVED STATUS** reads database state without starting a new scan. Monitoring admits at most one request every 15 minutes globally, with a normal 24-hour repository cadence. A configured signed GitHub push webhook invalidates changed-commit evidence and queues a bounded check. Default-branch changes/deletion also invalidate evidence; this does not guarantee scanning every intermediate push. Consented advisory checks refresh even on the same commit. GitHub user authorization and short-lived repository tokens renew automatically where the approved grant remains valid; reconnect if access is revoked or expired.
9. Open the individual report and compare SHA, last check, coverage, ruleset and each independent result: RUNCHECK RepoScan Static status/profile, dependency check/import context, RepoScan Intelligence consent/coverage and runtime host evidence. Its **OVERVIEW**, **EVIDENCE ENGINES**, **FINDINGS** and **COVERAGE + LIMITS** tabs open separate views via ?view=overview, ?view=evidence, ?view=findings and ?view=coverage; a direct URL keeps the report's existing public/private access rules. A changed commit or analysis profile needs new evidence. After 30 days without a commit (default threshold), an inactivity warning appears; after 48 hours without a confirmed check, monitoring is stale. Commit age is not a guaranteed record of every push, and neither warning proves malicious intent.
10. Every completed repository audit appears automatically in **REVIEW**, beside **AGENTS** and **TOKENS**. A private repository has a public summary; open its full report from **USER** → **MY CODE AUDITS** with the signed owner/admin wallet (?access=private). **WITHDRAW** stops future scans and revokes request access, but does not hide a completed outcome: its historical card remains **OUT OF DATE · MONITORING STOPPED**. An unfinished withdrawn request is not a completed audit. Use **REVOKE WALLET ACCESS** under **USER** → My GitHub connection (/user/github) for the wallet's GitHub connection; uninstall/restrict the App in GitHub to revoke access globally. Work already reading/transferring data may finish its bounded operation. Neither withdrawal nor semantic revocation retracts provider processing already authorized, screenshots or earlier report copies.

Opening **OPEN REPORT** from **USER** uses the signed detail (?access=private) for either repository type. This lets the owner/admin inspect queued requests and private request context without making them public. The ordinary report URL has no signed-detail parameter: completed public code reports are visible to everyone, and completed private-repository reports show only the redacted summary.

The repository outcome badges below summarize saved code-review evidence, separately from the native platform badge described above:

Interface outcome	Criterion and limit
PASSED	Exact commit and valid check within 48 hours, nonempty complete relevant-text coverage and static, Intelligence and dependency results all checked without limits. Findings are absent or informational only, including narrowly validated nonblocking contextual warnings. Applies to the displayed commit/scope, not a security guarantee.
NEEDS REVIEW	A warning-level source-code signal needs human investigation to assess actual code risks. Open its file/line and coverage; it is not proof that the agent fails to work.

Interface outcome	Criterion and limit
NOT PASSED	At least one critical source-code signal requires urgent investigation. This is not a failed runtime capability test or an automatic fraud accusation.
PARTIAL	Evidence is missing or coverage/analysis is limited, disabled, unavailable, not consented or malformed. Provider failures and unsupported/unscannable source mean incomplete evidence, not a code failure. Zero findings does not complete a partial check.
IN PROGRESS	No valid completed commit/check exists yet. Follow your request in USER; a queued request is not published as a completed report.
OUT OF DATE	The check is older than 48 hours: its previous result does not prove the repository's current head. A detected push separately invalidates old evidence.

A completed private-repository audit appears as a public outcome summary, never as public source or a full private report. Custom-build requests and request briefs remain private. Older `/reviews` links redirect to `/review`, retaining compatible filters. A custom request, code report and capability certificate remain different documents.

Report signal	How to read it
Queued / access pending / scanning	No completed current result yet. Resolve access or wait for the bounded queue.
Limited static review, no rule matches	No selected heuristic matched the analyzed files. Check coverage: this is not a security clearance or a "NON LARP" guarantee. Configuration-only, fixture-only and unsupported repositories do not receive this result as evidence of a working agent.
RUNCHECK RepoScan Static checked / partial / unavailable / disabled	Checked means only that pinned static/offline analysis completed for the displayed bounded snapshot. Partial, unavailable or disabled never means clean, and zero retained findings is not a safe-to-install or "NON LARP" verdict. Read the exact engine revision and file counts.
Amber attention / partial coverage	Read the findings and unscanned scope. Ask for deeper review rather than assuming the rest is safe.
Red critical rule match	Inspect the referenced file/line and correct or explain the issue. A pattern match requires investigation, not a public fraud allegation.
Changed commit / unavailable / stale	Prior evidence must not be interpreted as a current clearance. Read the latest status and check time.
Dependency lookup off / unavailable / partial	No full dependency verdict exists. Off means you did not authorize OSV lookup; unavailable means the provider could not be checked; partial means versions or manifests fell outside supported coverage.
Dependency advisory found	Compare the exact locked version and advisory ID. Direct/transitive/unknown context and observed import references help investigation but do not prove the loaded version, runtime reachability or exploitability. No reference found is not a safety clearance.
Semantic disabled / not consented / consent stale / partial / unavailable	No completed current semantic result exists. Check configuration or the disclosed provider/profile and consent; offline findings remain separate.
Semantic findings	Fixed-category AI inferences attached to file/line references, requiring human review. They are not verified vulnerabilities or a "NON LARP" verdict.
Runtime host-attested / not-attested	Current host evidence links a running supported OCI image to the reviewed SHA and a pinned builder workflow. Missing/expired evidence is not current verification; the trusted host and routing remain assumptions.

What the automatic checks examine

Version repository-static-v8-ast-resumable checks the selected branch's exact commit and rechecks its head before accepting the result; an approved commit pin must match. It looks for exposed credentials, dangerous execution, disabled TLS validation, sensitive logging, unsafe deserialization, installation hooks and risky workflow permissions. Contextual review retains the original rule/severity/location and explains narrowly validated informational, nonblocking warnings, such as passive configuration documentation or fixed literal-only test patterns. It is not a repository/owner whitelist. Actual secrets, unsafe TLS, exfiltration, arbitrary execution, instruction overrides and advisory risks remain for review; malformed saved context makes evidence partial. No matched secret value or source snippet enters a finding.

The base scanner analyzes up to **1,000 eligible text files, 1 MiB per file, and 16 MiB of source in total**, within 95 seconds. Archive limits are 20 MiB compressed, 64 MiB expanded and 20,000 entries. Supported source/configuration/documentation/dependency text defines relevant coverage; vendor/generated files, binaries, unsupported formats, links and submodules are declared exclusions, not automatically incomplete relevant-text coverage. Oversized eligible text, resource skips, unfinished batches and truncated findings are partial. At most 250 prioritized base findings are retained. No submitted code, dependency, script or hook is executed or installed; complete never means every repository byte or proven cross-batch relationships.

When enabled, RUNCHECK RepoScan Static runs the pinned reposcan-static-v1 engine with adapter-v4-resumable-shards. Deterministic sequential shards target **32 files / 256 KiB source**. These soft targets do not discard larger accepted files: each stays whole in its own shard. The hard ceilings remain **200 files / 2 MiB source / 3 MiB JSON per invocation, 1 MiB per file, and 1,000 files / 16 MiB source overall**. All shards share 75 seconds inside the 95-second parent budget; at most 100 aggregate prioritized findings are retained, with truncation partial. The static graph keeps its 60-second internal limit and hard resource ceilings. It is static-offline only: no model-source transfer or internal advisory lookup. OSV remains the independent optional choice in step 5.

Private, source-free checkpoints can reuse completed offline shards during a later admitted check only for the same exact commit, analysis profile, verified adapter, input snapshot and shard layout. Their **24-hour TTL** starts with the continuation; retrying does not extend it. Unattempted shards are prioritized before incomplete retries. Checkpoints contain validated rule/file/line metadata, counts, check times and fixed limitation codes, not code, snippets or engine prose; they stay private and outside the sandbox. Read reused/completed shards, current-run file counts and continuation expiry in coverage. Invalid, expired or storage-limited continuation is not a clean result. The normal **24-hour repository cadence** and **15-minute global admission budget** do not become faster. Large repositories remain partial when any required pass exceeds its limits; reuse never forces **PASSED** or guarantees full or cross-shard analysis.

The 1 MiB per-file allowance is local/offline only. External AI source transfer remains separately consented and limited to 64 KiB per file; its other limits are unchanged. Files that cannot fit a batch's JSON request after escaping are explicitly omitted for budget, never counted as inspected.

The selected snapshot enters a credential-free Bubblewrap sandbox with no network, no host checkout/home/secret mounts and no inherited service environment. Files exist only in sandbox tmpfs while the child runs and disappear when it exits; only validated rule IDs, fixed explanations and repository-relative file/line locations are retained. The existing accepted-request, periodic-check and signed-webhook queue invokes this automatically. Opening or refreshing a report does not run it, the one-review-per-15-minutes global budget still applies, and a changed pinned engine profile invalidates the old cached result.

The optional advisory check reads exact npm manifest/lock v1–v3/shrinkwrap pins, pnpm 5/6/9, Yarn classic/Berry, Python requirements/poetry/uv, registry Cargo, Go module/sum and Composer manifest/lock records. It sends one batch of at most 100 unique coordinates, with a five-second deadline. Unsupported syntax, ranges, private/local/git sources, replacements and oversized or excess files are explicitly incomplete; no dependency gets installed. A Go checksum file can include unused historical versions. Known advisory matches retain direct/transitive/unknown context. Local import/require/namespace checks cover JS/TS, Python, Go and Rust, distinguishing source from test paths. Dynamic imports, renamed modules, PHP namespace associations and unobserved usage stay unknown; imports do not resolve exact loaded versions or prove exploitability. Provider failure is unavailable, never zero vulnerabilities.

The local dependency parser also admits 1 MiB per file and 16 MiB overall, with unchanged limits of 10,000 records and 100 retained coordinates. Limit hits remain explicit incomplete coverage; external advisory queries still require separate consent.

Immediately before sending the OSV batch, RUNCHECK re-reads the saved owner's consent and the current scanning request revision from the database under the lookup deadline. A stale checkbox, revoked consent, changed request or failed authorization cannot authorize the transfer or republish the superseded combined result. No consent setting overrides OSV's 100-coordinate ceiling or the independent AI v2 limits below.

Source is transient in RUNCHECK: the base scanner uses memory and RUNCHECK RepoScan Static uses only bounded sandbox tmpfs, removed on exit. Separately consented RepoScan Intelligence sends its limited redacted selection to the access-controlled RUNCHECK analysis gateway. RUNCHECK stores metadata, consent profiles, commit, coverage and fixed-message findings, plus the private source-free offline continuation subset described above. It does not store raw source, snippets, unfiltered tool output, engine prose or stderr. GitHub user grants are encrypted and access-bound; installation tokens remain temporary. Public reports expose displayed metadata and finding paths, not your brief. The handling statement is not a signed NDA or code escrow; agree extra confidentiality obligations before granting sensitive access.

Private GitHub access, semantic review and runtime evidence

- **GitHub access:** after installation, confirm the exact selected repositories. The service renews expiring user grants and rechecks authorization before scans; no manual reconnect is needed merely because 30 minutes passed. Reconnect if GitHub access/refresh authorization is revoked/expired or the operator changes the signing key. Adding a repository in GitHub does not silently authorize it for your wallet: reconnect to verify access. Operator App registration/configuration is a human prerequisite, not a missing implementation.
- **Semantic review:** consent v2 requires a fresh checkbox for the current profile and is rechecked before each deterministic batch. Standard profile limits: **8 batches, 64 files per batch, 64 KiB per file, 128 KiB source per batch, 1 MiB source overall**, one shared **15-second** deadline and **8,192 aggregate output tokens**. At most 32 aggregate prioritized signals are retained; each response is limited to 64 KiB. Read eligible/submitted/inspected/skipped/redacted counts and completed/planned batches. A failed later batch retains earlier findings but means partial, never clean. Known secret patterns are redacted, not all confidential details; the model has no tools/actions. Read the actual profile offered before authorizing it. Profile changes require renewed consent; revocation stops future authorized transfers.
- **Runtime evidence:** ask the independent deployment-host operator to connect the reviewed SHA to a running, digest-pinned read-only OCI image with verified GitHub Sigstore/SLSA build provenance and a separately approved builder workflow/commit. The host signs short-lived observations; normal evidence lasts ten minutes, the maximum accepted window is 15 minutes. Check the displayed SHA, image digest, origin and expiry. Host-attested trusts that host and its routing; it is not hardware attestation, proof of safe behavior or a substitute for the agent's live capability test. A new SHA or expired observation requires new evidence.

What remains yours: select/authorize GitHub repositories, decide whether each optional disclosure is acceptable, arrange actual host/deployment evidence where desired and review findings with the developer. Missing provider/App/host configuration is shown as unavailable rather than invented success.

Example: reviewing an external support agent

Submit the repository URL first with a brief such as "This agent answers support questions through Telegram; review credential handling, model calls and deployment workflows." RUNCHECK reuses valid wallet-linked access for that exact private repository. Only if authorization is missing or needs renewal, connect the GitHub App during the request and then verify/queue the saved request. Inspect existing connections in USER, not in the new-request form. Enable OSV only if sharing package identifiers is acceptable, and semantic review only if the displayed provider may receive the selected source. Review each commit-specific amber/red signal with the developer, push a correction and await the new report. Ask the deployment host for current runtime evidence if you need source-to-image linkage; verify the live Telegram capability separately. None of these reports creates a RUNCHECK agent or grants a token-launch certificate.

Publish, unpublish and inspect controls

1. From the ready agent's **AI + SERVICES** view, open AI + deployment setup (stage=model) and use **PUBLIC DIRECTORY** to publish its profile. Open it signed out and confirm the title, description and certificate link are the intended public data.
2. Unpublish from the same control to remove the listing. Runtime execution, on-chain tokens and existing proof records are not reversed.
3. In Advanced Settings, **RUN MANIFEST CONTROL CHECK** checks the saved configuration; **VERIFY PERSISTED DIGEST** → checks the stored record's digest. Neither upgrades a capability result. Credential references are states, not values to copy into chat.
4. Before token launch, inspect every launch-gate row: correct owner, current factory runtime and its automatic model-response certificate, accepted current terms and verified Pons configuration. Resolve the named row rather than repeating signatures.



RUNCHECK_

END-USER GUIDE · 15

14. Runcheck Agent and manual order controls

These controls belong to the shared **Runcheck Agent** at /full. You do not need to create an agent. The **Personal Companion** template provides its own AI web chat, not these manual-trading or Telegram notification panels. An optional shared Telegram connection is available here using RUNCHECK's bot; follow section 19 without creating a Telegram Assistant.

Conversation, chart and voice

1. Open Runcheck Agent: you land directly in the conversation. Type a question or open **PROMPTS** and choose from 18 suggestions in six groups, edit the draft and choose **SEND**. Tools scroll independently beside the chat on desktop; use **TOOLS** then **Markets** on mobile. Choose the market and its venue where offered, then 1h, 4h or 1d. Check the analyzed asset identity and observation time; a stale chart is not a live execution quote. Missing history stays unavailable, and reference-price samples are not real OHLC candles.
2. Ask a market question. Sign in for private portfolio/position context; confirm that the account displayed is yours.
3. For a structured proposal, read its rationale and choose **APPLY TRADE SETUP TO LIGHTER PANEL** when present. This copies fields only. Ordinary prose without this control is not an executable setup.
4. Use the microphone control, allow browser permission, record and stop. Review/edit the transcript before sending, or cancel it. Playback controls read a response; pause/resume/stop do not affect trading. If voice is unavailable, type instead.
5. On a narrow screen, switch between **CHAT** and **TOOLS**. Inside **TOOLS** choose **Markets, Vaults & auto** or **Telegram**; the wallet and review requirements are unchanged.
6. Read older messages with the transcript's own scroll; **Latest messages** returns to the latest response. **COPY** copies the displayed answer. Expand **Sources** beneath an answer when available to inspect its supporting links; copied text and voice do not replace source review. A draft can be edited while waiting; sending waits for the current response to finish.
7. If the panel says **RESEARCH ONLY**, do not expect an opening-order preview. Exact-contract or other non-executable analyses do not substitute a same-symbol perpetual. Existing positions, reduce-only exits and recovery remain available; explicitly select an executable perpetual for a new order.

Manual perpetual order, end to end

1. Fund/select the correct **ACCOUNT BOUNDARY** as in section 10. Review integrator identity, fee limit and **APPROVAL EXPIRY**.

2. Decide whether to retain the local key in this tab's session storage. With that option off, it stays in memory. Select **ENROLL LOCAL SIGNER**, and **VERIFY ENROLLMENT** if the panel requires verification.
3. Fill the following fields, then **BUILD IMMUTABLE PREVIEW**. No order is sent at this step.

Field	Meaning / unit
DIRECTION	LONG or SHORT for the selected perpetual market.
NOTIONAL USD	Gross position size in USD, not the collateral deposit.
LEVERAGE	Requested leverage, 1–20 in the form, subject to venue checks.
MAX SLIPPAGE BPS	1–500 basis points; 100 bps = 1%.
STOP LOSS / TAKE PROFIT	Absolute market prices. Do not enter percentage distances here.

4. Read the immutable entry and protection review, amount, account, fees and confirmation text. Use **SIGN AND SUBMIT EXACT PREVIEW** only for that reviewed operation.
5. Inspect the resulting venue status and positions/protection. If pending or unknown, use **RECONCILE** in **PENDING / RECOVERABLE RECONCILIATION** before another order.
6. To exit, select **REVIEW CLOSE**, inspect its reduce-only **EMERGENCY CLOSE** preview, then sign. **CANCEL CLOSE** dismisses only the review. Check remaining exchange orders at the venue and cancel any no-longer-needed orders there.
7. **CLEAR TAB SIGNER** removes the local key, not positions or venue permission. Use the venue's key-management flow for actual revocation. This browser key is separate from the server Autotrader key.

The browser saves a recovery record before either leverage or order submission. This record contains public operation metadata, not your signing key, and remains after a reload. Reconnect the same owner/account and unlock its browser signer, then select **RECOVER SAVED MUTATION**. Recovery never resends an order. A leverage-only attempt resolves after its signing-key transaction number advances or its signed message expires; an order remains blocked until independent reconciliation verifies its final outcome. An HTTP success or an “acknowledged/pending” response is not final confirmation.

Manual 0x spot swap, end to end

1. Choose **SELL TOKEN**, **BUY TOKEN**, **SELL AMOUNT** in human token units and **MAX SLIPPAGE · BPS** (1–1000). Only reviewed tokens/routes are offered. For USDG/WETH, **INVERT PAIR** exchanges the selections for USDG → WETH or WETH → USDG when both are permitted; identical sell/buy tokens are not allowed. Changing the sell token or inverting clears the amount and old price/quote. Re-enter an amount in USDG or WETH units as appropriate; the same number does not represent the same value in both tokens.
2. **CHECK INDICATIVE PRICE** is informational. Select **REQUEST FIRM QUOTE** for an actionable wallet-bound review.
3. Inspect output/minimum received, fee recipient, quote expiry, spender and swap target. If **RUNCHECK** charges a fee, it is in the sell token: USDG on a USDG sale and WETH on a WETH sale; zero **RUNCHECK** fee does not remove gas/other costs. If shown, **APPROVE EXACT ...** approves the required sell-token amount; wait for confirmation and the fresh firm quote afterwards. USDG approval does not authorize WETH. Review each direction independently; a configured route may still have no usable liquidity.
4. Select **SUBMIT REVIEWED SWAP** and confirm the exact wallet transaction. Preserve its hash until two confirmations and minimum received have been checked.
5. Reopen the same wallet panel to recover a pending intent. A reverted transaction and a conflicting reconciliation are different outcomes; a conflict requires investigation, not another swap. Spot has **NO NATIVE TP / SL**.

15. Autotrader field reference and recovery

Autotrader activation authorizes RUNCHECK to receive a dedicated Lighter API private key and store it encrypted in the operational database. The worker uses it to sign the orders permitted by your policy. Manual trading keeps its key in the browser instead. Neither flow needs your wallet private key. **STOP & DESTROY WORKER KEY** removes the active stored credential; history and protected backups have separate retention. Venue-level invalidation requires key rotation or revocation in Lighter. See the [Privacy policy](#) and [Terms of service](#).

Prerequisites: a ready factory Autotrader, the owner session and a funded, exclusively dedicated Robinhood Lighter account index. Paper verification and chat suggestions do not activate the worker. Open its Autotrader view in USER or select it in Runcheck Agent's Vaults & auto panel. Fixed rules and Hourly AI strategy are separate modes; choosing a tab cannot migrate or activate an existing delegation.

Fixed rules: request and apply advice

This advisor uses deterministic rules, not an AI model. Existing single-market delegations keep their original settings and signed messages. Each retains its signed interval; the hourly AI mode cannot silently replace it.

1. Open USER → **MY AGENTS**, select the trader, then its **AUTOTRADER** view and enter the dedicated account index.
2. Enter **Market to analyse** and **Analysis and execution timeframe**. Supply **Maximum allocated collateral USD**, **Estimated loss budget per trade USD**, daily/total entry-turnover limits, account drawdown entry-stop, maximum leverage and authorization duration.
3. Choose **RECOMMEND STRATEGY & SETUP**. Inspect the observed market/account time, rationale, size and protection distances. A waiting/blocked result has no applicable policy; read its reason.
4. Inspect **HISTORICAL SETUP SIMULATION** when available. It describes the simulated sample, not a future return or guaranteed loss ceiling. Reference entry/SL/TP prices are not resting limit orders.
5. Use **APPLY SETUP · DOES NOT START TRADING** while advice is valid. Advice expires after five minutes; changed inputs need a fresh recommendation. Editing the applied policy makes it a custom setup.

Fixed rules: review every policy field

Field	Interpretation
Perpetual symbol	Supported market such as BTC; not a spot-token contract.
Strategy	dca, momentum, mean - reversion. Rebalance/range paper proposals are not real worker strategies.
Direction	LONG or SHORT, within the reviewed strategy.
Execution candle timeframe	1m, 5m, 15m, 1h or 4h. Closed-candle data used for signals.
Closed-candle lookback	20–100 candles; not seconds.
Signal threshold bps	10–1000; deviation required by signal strategies.
Maximum entry notional USD	Maximum gross size of one entry.
Daily / Total entry turnover USD	Cumulative entry notional budgets; not available collateral or PnL.
Equity drawdown stop USD	Blocks new entries after the configured equity loss; does not flatten the account or guarantee the final loss.
Leverage	1–20 in the form, further constrained by venue/account checks.

Field	Interpretation
Maximum slippage bps	Relative tolerance: 100 bps = 1%.
Stop loss bps / Take profit bps	Distances from entry, unlike the manual panel's absolute prices. For example, 200 bps = 2%; this is a unit example, not a recommended setup.
Entry evaluation interval seconds	At least 60. How often entry may be evaluated; distinct from candle timeframe and not a promise of a trade each interval.
Delegation duration hours	1–720; expired authority cannot start new trading.

Hourly AI strategy: preview before authorization

1. Choose **Hourly AI strategy**. Add 1–8 exact markets from the verified Robinhood Lighter catalogue, then choose permitted LONG/SHORT directions. The model cannot add markets outside this signed list.
2. Set the entry notional, estimated loss per trade, daily/total entry turnover, account drawdown entry-stop, leverage, slippage and authorization duration. Initial values are editable examples, not a personalized recommendation.
3. Expand **Hourly scheduling and analysis limits** to review setup validity, selection/model-call caps and market cooldown. Evaluation is once per hour, not one guaranteed order per hour. A pending setup or existing position blocks overlapping entries.
4. Choose **Preview one AI cycle**. This sends only the selected policy for public-market analysis, not wallet keys, balances or private portfolio data. It neither registers a key nor places an order. Analysis can take two minutes; the agent allows one preview per two minutes and 24 per UTC day. Missing data, disabled inference or an invalid result blocks this path, without switching to fixed rules.
5. Inspect HOLD or the selected exact market, direction and original entry/stop-loss/take-profit prices. Preview completion time and market-data observation time are separate. Changed inputs or an expired preview require a new preview. HOLD is a valid no-entry result, not a failed transaction.
6. If you choose to authorize future cycles, enter the dedicated account and tick its explicit real-trading consent. **Register dedicated key and review AI delegation** requests the separate venue-key registration and shows the exact v2 policy. It does not reuse the preview as an order.
7. Review the new delegation, then use **SIGN DELEGATION & ACTIVATE REAL TRADING** only if you accept unattended real orders within it. The worker performs fresh analysis. Without this final owner signature, no strategy starts.

AI field	Boundary
Allowed markets / directions	1–8 exact venue markets; LONG, SHORT or both. No arbitrary token contracts.
Entry notional / turnover	Entry notional must not exceed the daily budget, which must not exceed the total budget. Budget is reserved against the UTC day of reservation, not the eventual fill time.
Estimated loss limit per trade	Must not exceed the account drawdown entry-stop. Estimates include protection distance and cost allowance; realized loss is not guaranteed.
Maximum leverage / slippage	1–20× and 1–100 bps, further constrained by venue checks.
Authorization duration	1–720 hours; replacement requires a new reviewed signature.
Setup validity	60–14,400 seconds. Original levels remain fixed while waiting.
Selection / model-call limits	Each 1–24 per UTC day. Model calls and selected setups are counted separately from fills.
Market cooldown	3,600–86,400 seconds before selecting that market again.

Activate, monitor, stop and reconfigure

Paused Autotrader controls and strategy-advisor fields. Static DEMO data, not a recommended strategy; no real account or operation is active.

- RUNCHECK · End users · 18 October 2026 · Europe/Rome

6. **STOP & DESTROY WORKER KEY** stops worker signing locally. Rotate/revoke that API key at the venue as well for venue-level revocation, then resolve remaining positions/orders.
7. If offered, **VERIFY FLAT ACCOUNT & ALLOW RECONFIGURATION** performs fresh checks after old signed packets expire. It requires a cleared account and does not reactivate anything. Obtain fresh advice, review and sign a new delegation to change the active policy.

The policy limits are enforced by RUNCHECK's worker; they are not native cryptographic limits on every use of the Lighter API key. This is why the dedicated account and explicit venue revocation matter. New advice never silently changes a running delegation.

16. Vault recipes and permission boundaries

Create and move funds

1. Open **USER** → **MY AGENTS**, select the agent, then its **VAULT** view (**AGENT VAULT**). Check the owner and chain. If absent, supply a separate guardian address and **REVIEW VAULT CREATION**.
2. Review destination/value and **CONFIRM IN WALLET**. Wait for 12 confirmations; record the deployed vault address from the panel, not an example address.
3. Enter an **ETH amount**, **REVIEW DEPOSIT**, inspect and confirm. After confirmation, compare wallet/vault balances. To reverse the test, enter the intended withdrawal amount and **REVIEW WITHDRAWAL TO OWNER**, then confirm and recheck.
4. For ERC20, expand **ERC20 TOKENS · DEPOSIT AND RECOVERY**, enter the exact contract and **CHECK TOKEN BALANCE**. Verify decimals/symbol and owner/vault balances.
5. Enter **Token amount** in human units. **REVIEW TOKEN DEPOSIT** transfers from owner to vault; **REVIEW TOKEN WITHDRAWAL TO OWNER** recovers to the owner. Review and sign each separately.
6. A review expires after 120 seconds. Prepare a new one if it expires before signing. If already broadcast, retain/reconcile its hash instead of preparing another transfer.

Token and session inspection starts immediately when requested. Changing the contract, session, recipient or amount clears the previous review: wait for the matching result and prepare again. A balance from the previously inspected token cannot authorize the new token. Before signing, select the owner extension if prompted; RUNCHECK rechecks the account, mainnet, expiry and transaction simulation. An ERC20 simulation returning failure blocks submission.

Grant and revoke a native-ETH session

1. Expand **SESSION KEYS · LIMITS AND REVOCATION**. Supply a separate **Session wallet**, then **CHECK ON-CHAIN SESSION**.
2. Set **Duration (hours, 1–168)** and positive **ETH per call**, **ETH per day**, **ETH total** limits. Keep per-call within daily, and daily within total. **REVIEW SESSION LIMITS** → inspect → **CONFIRM IN WALLET**.
3. Enter the intended **Allowed destination wallet**. Review/sign **1 · REVIEW TARGET PERMISSION**, wait for confirmation, then separately **2 · REVIEW RECIPIENT PERMISSION**. A configured session alone has no permitted destination.
4. Recheck active state, expiry, spent amounts, nonce and generation. This panel grants permissions; it does not submit a transfer signed by the session wallet.
5. **REMOVE RECIPIENT** removes one destination permission. **REVIEW SESSION REVOCATION** revokes the session itself. Confirm and reread the on-chain state.
6. To suspend sessions across the vault, **REVIEW PAUSE** → confirm. **REVIEW RESUME** restores eligible permissions, not revoked/expired ones.

Role / control	Can do	Does not do
Owner wallet	Create/manage vault, deposit, recover to owner, manage sessions and pause/resume through this panel.	Automatically delegate Lighter or transfer ownership by switching login.
Separate guardian	Emergency pause authority in the vault contract.	Withdraw owner funds; log into the owner-only panel as the owner. Guardian contract interaction is a separate emergency path.
Session wallet	Only allowed on-chain session operations within its permissions, expiry and budgets.	Unlimited withdrawals, ERC20 trading authorization, or Lighter signing merely because a session exists.
Archive agent	Remove an eligible agent from normal lists.	Delete its vault, balances or on-chain history.

For an archived agent, open **MY AGENTS** (/user/agents) → **ARCHIVED AGENTS · RECOVERY** → **VAULT RECOVERY** → , select it with the original owner wallet and use the retained withdrawal/revocation controls. This exact-agent workspace is recovery-only. Archiving is not a substitute for retiring workers or recovering capital.

DOCUMENTATION PREVIEW · STATIC EXAMPLES · NO TRANSACTIONS

– AGENT VAULT · OWNER-CONTROLLED CAPITAL

Dedicated on-chain funds for this agent. Your wallet signs every action. A vault does not automatically enable trading.

CHAIN 4663 · NO VAULT CREATED · 12 CONFIRMATIONS

Create one vault for this agent. The owner controls withdrawals; a separate guardian can only stop session execution. Gas is paid in ETH.

Separate guardian wallet

0x...

REVIEW VAULT CREATION

Vault creation control. DEMO identity only: no vault was deployed and no displayed address is a funding destination.

17. Token launch and operation recipes

Launch form and exact signature

1. Open /launch, the token-only launch console, and select the eligible factory agent. Create an agent in /launchpad first if necessary. Complete every launch-gate check, including **ACCEPT CURRENT TERMS** when required by the current interface. An external-only agent remains proof-only.
2. Enter **NAME** (V2: up to 60 characters), **TICKER** (V2: up to 20; uppercased), and **DESCRIPTION** (up to 256). V1 retains its 32/10 limits. For testing use “Demo Agent Test”, not an official brand launch.
3. Choose **TOKEN IMAGE**: PNG, JPEG, WebP or GIF, maximum 5 MB. A local preview appears immediately. Confirm public storage, then **UPLOAD / RETRY SELECTED IMAGE**; subsequent selections upload automatically while consent is checked. Yellow means pending confirmation/upload; red means the upload failed; green **IMAGE SAVED** means it completed. Only then is **OPEN PUBLIC IMAGE** available. RUNCHECK stores immutable image bytes in its database with backups and a stable public HTTPS URL, not IPFS. Upload failure retains the selected file and preview; retry without recreating the agent.
4. Use **SEARCH PAIRED ASSETS** to search by symbol or contract address, then select **PAIRED ASSET**. The list includes native ETH and all currently discoverable, approved Pons quote tokens, not just USDG. Contract

links distinguish similar symbols. This selects the asset paired with your new token; an existing pool is not selected. The display catalogue is cached for up to five minutes; approval and economics are freshly rechecked before signing. Add optional X/Telegram profiles and enter **DEVELOPER BUY** in the selected asset: 1 means 1 ETH for an ETH pair or 1 USDG for a USDG pair. Changing the asset resets the buy to zero. Enter 0 for no initial buy. You still need the amount plus the launch fee and gas (always paid in ETH). Excess fractional precision is rejected, never rounded.

5. Open **ADVANCED. CREATOR TAX (%)** and its slider show percentages: 7 means 7%, not 7 basis points. The current maximum and temporary launch-window tax come from Pons. **SNIPER TAX EXEMPTIONS** accepts up to 32 distinct, non-zero full wallet addresses, one per line or comma-separated. Creator and creator-fee wallets are automatically exempt from that temporary tax, not ordinary fees. Optional buyback/vesting allocates future fees to locked tokens; it is off by default and is not a cash payout or holder fee sharing.
6. **CREATOR WALLET** may be left blank to use your connected wallet, or set to a different recipient for creator fees. It does not change the paying wallet/token owner. Choose whether to publish the agent profile. Review the live estimated launch cost. Holder fee sharing is a separate Pons post-launch flow with two more wallet confirmations (distributor deployment and fee routing); it is not automatically enabled in this form.
7. **LAUNCH TOKEN · REVIEW TRANSACTION** performs fresh verification and simulation; it does not send the launch. The button is visible above and below the form, with missing requirements listed below. Use **REFRESH READINESS** if needed. For an ERC20 initial buy, **APPROVE EXACT AMOUNT IN WALLET** may first request an allowance reset and/or exact approval. Wait and prepare again afterwards.
8. Inspect **FROM / CREATOR**, Pons **V2** factory/launch-and-buy router, ETH launch fee, initial buy in its own asset, total ETH value, fee wallet, fee split, predicted token/curve and graduation threshold. Read the live economics, not a legacy fixed percentage.
9. Tick the exact-transaction review confirmation and **SIGN + LAUNCH ON PONS**. A signed site session and a browser-extension connection are separate. If prompted, select the extension holding the verified owner wallet, unlock it and approve connection. Multiple installed wallets require a selection; the site never silently chooses a different account. Select Robinhood Chain mainnet (4663). For Phantom, enable the network in the extension if unavailable. The browser verifies the token, pool and minimum output before requesting the transaction. Cancelling wallet selection sends nothing. Preserve the returned hash and wait for **ON-CHAIN LAUNCH VERIFIED**. A prepared transaction or connection approval is not a launched token.
10. On reload, inspect the recovered pending launch and use its reconciliation controls. Follow the resulting token link only after verification; do not relaunch to fix slow indexing.

Before the wallet can send, the server reserves this launch and its transaction number. The reservation remains active even if the wallet returns no hash or the browser loses its response; an expired review cannot create another launch while this reservation is active. Copy the original or replacement hash from wallet activity into the recovery field and use **CHECK CONFIRMATION**. The exact transaction and its on-chain outcome must be verified. **RELEASE UNUSED RESERVATION** appears only when the saved request never reached the wallet-send step; it is not a way to cancel a broadcast transaction. Never sign a second launch to resolve an uncertain first one.

An operator's **ADOPTION PENDING** for an existing Pons token is a different flow: no browser wallet send or native confirmation is requested. Do not use another token launch to resolve it; follow **CUSTOM AGENT** in USER and contact RUNCHECK if it remains pending.

Shared procedure for Pons V2 operations

Open the token from Tokens, verify its contract/quote asset and expand **PONS V2 OPERATIONS**. Select an action below, **PREPARE EXACT TX**, read the fresh review and **SIGN IN WALLET**. If an approval is prepared, complete it and prepare the actual operation again. Preserve the hash in **RECENT OPERATIONS**; check the receipt and refreshed balances/phase before another action. Never force a disabled action with a different wallet or stale review.

The review displays amounts in their token units and identifies the asset contract; unknown metadata is labelled unavailable, never guessed to be ETH. Editing action, amount, slippage, recipient or buyback invalidates the old review. If several wallet extensions are installed, choose the one holding the signed-in owner account. Network

selection is verified after switching, including after adding a network.

Only one in-flight Pons operation is allowed per wallet, across token pages. Once the wallet request is opened, use **RECOVER / CHECK TRANSACTION** if a response is lost or the page reloads; do not start another trade. If needed, copy the hash from wallet activity into **TRANSACTION HASH FROM WALLET ACTIVITY**. A corrected/replacement hash must correspond to the same reserved wallet transaction, not an unrelated transfer. **RELEASE UNSENT WALLET REQUEST** is available only for a request not yet armed for sending. An uncertain wallet result remains blocked for recovery; signing in again does not prove the transaction was unsent.

If you cancel or replace the transaction from the wallet, recover the replacement hash. **RUNCHECK** checks the original owner, mainnet, reserved transaction number and 12 confirmations. **REPLACED** means a different transaction consumed that number: the original Pons operation was not executed. A pending replacement does not unlock another operation yet.

BUY and SELL

1. **BUY:** enter **AMOUNT** in the displayed, verified quote asset (for example ETH, USDG, WETH or cbBTC), set **SLIPPAGE BPS** (1–500, equivalent to 0.01–5%), prepare and review minimum tokens/curve-or-pool route. Complete any exact quote-token approval, prepare again and sign the buy. Success: confirmed receipt and increased token holdings.
2. **SELL:** enter the token quantity, not ETH proceeds; set slippage, prepare, complete any exact token approval, prepare again and sign. Success: decreased token holdings and credited quote asset. Wallet ETH remains necessary for gas.

SWEEP FEES and CLAIM QUOTE

1. **SWEEP FEES:** connect the current creator wallet, inspect accrued/distributable fees, select sweep and prepare. If eligible, sign and inspect the resulting escrow credit. If conversion/buyback requires the external Pons operator, stop and report that condition; a user signature cannot replace that operator.
2. **CLAIM QUOTE:** connect the beneficiary with positive **ESCROW QUOTE**, select claim, prepare and sign. No amount field is required. Success: the escrow debit and wallet credit correspond. Unswept fees must be distributed first.

RELEASE VEST and CLAIM VESTED

1. **RELEASE VEST:** inspect **VEST RELEASABLE**. When positive, select release, prepare and sign. Success: available vested tokens move into beneficiary escrow, not directly to the wallet.
2. **CLAIM VESTED:** connect that beneficiary, check **ESCROW TOKEN**, prepare and sign the claim. Success: wallet token credit and reduced escrow. If nothing is releasable/credited, wait for vesting or the prerequisite release; do not repeat an empty claim.

BUYBACK TOGGLE and TRANSFER FEES

1. **BUYBACK TOGGLE:** as current fee recipient, select the action, set **Enable future creator-share buyback**, prepare the intended enable/disable value and sign. Check the refreshed flag. Existing earmarked buyback balances are not converted back to cash by this toggle.
2. **TRANSFER FEES:** as current recipient, enter **NEW FEE RECIPIENT**, compare the entire address, prepare and sign. Confirm the new recipient. This transfers future creator fees and vest-beneficiary rights; existing escrow remains with its prior beneficiary. The previous wallet cannot simply toggle the new owner's fees afterwards.

GRADUATE and CREATE V4 POOL

1. **GRADUATE:** inspect curve reserves/threshold and phase. Only when eligible, prepare and sign graduation. Check the confirmed new phase; it may still require pool creation.
2. **CREATE V4 POOL:** when the post-graduation phase permits it, prepare and sign pool creation. Check the pool/phase and refreshed trading route. These controls do not ask you to fund additional LP liquidity. During a

transition or anti-snipe window, wait for the displayed on-chain condition before retrying.

18. X publishing and scheduling reference

Publish one reviewed action

1. Open your ready Social agent, confirm its connected X handle and publishing service/consent. X identity proof alone does not authorize posting.
2. Choose **NEW POST**, **REPLY** or **QUOTE POST**. Replies/quotes need **TARGET POST ID**, the numeric part of the post URL, not the whole URL.
3. Enter the text under **PUBLISH WITH THIS AGENT**. RUNCHECK applies its displayed weighted-length limit (280) and at most one cashtag. Inspect the target, content and current cost before **PUBLISH ON X**.
4. On success, **OPEN VERIFIED X POST** ↗ and check author/text/target. If uncertain, **CHECK X STATUS** first. **RETRY SAME X INTENT** is only for a safely unsent intent; **X INTENT LOCKED** means a provider attempt cannot be blindly repeated.
5. Use **START NEW X INTENT** only when the interface has authoritatively established the previous attempt was absent/rejected. Do not clear storage to bypass duplicate protection.

Configure the engagement worker

Start paused, fill the following controls and **SAVE ENGAGEMENT WORKER**. Then **RUN DRY TEST (READ + MODEL COST)**: it may incur data/model cost but does not publish. Inspect its decision before enabling the schedule.

Field	What to enter / inspect
AUTOMATION	Paused for setup; Enabled plus save permits scheduled real publishing when the release gate is open.
POSTS PER DAY	Hosted/Managed: 1–15. BYOK: displayed numeric cap or NO RUNCHECK DAILY CAP ; provider limits still apply.
READ FOLLOWED FEED EVERY	Select 30–720 minutes from the offered intervals.
POSTS PER FEED READ	1–100; larger batches increase reads/cost.
ENGAGEMENTS PER FEED READ	Fixed one per feed read; not one per fetched post.
MINIMUM ACTION SPACING	5–1440 minutes between actions.
LANGUAGE	Desired language/locale for generated content.
REPLY WHEN RELEVANT	Whether relevant feed items may receive replies.
NATIVE QUOTE WHEN AVAILABLE	Quote only through available provider capability.
ALLOW SELF-SERVE URL QUOTES	Separate opt-in to the displayed potentially higher-cost fallback.
TOPICS / BLOCKED TOPICS	Comma-separated desired/excluded subjects.
CHARACTER	Persona/instructions, within the displayed 4000-character limit.
KNOWLEDGE	One factual instruction/rule per line; never secrets.

Field	What to enter / inspect
OFFICIAL SOURCES	One named HTTPS source per line, in the format below.
ACTIVE FROM / ACTIVE UNTIL	Daily activity window, interpreted in TIME ZONE.
TIME ZONE	Valid IANA zone, for example Europe/Rome; not a free-form city label.

The OFFICIAL SOURCES format is NAME | HTTPS URL, for example RUNCHECK | <https://runcheck.bond/docs>.

1. After reviewing the dry test, set Enabled and save. **RUN ONE LIVE CYCLE** immediately evaluates a live cycle when permitted; it may publish and is not another dry test.
2. Inspect next post/feed times, post/reply/quote counters, feed reads, skipped/failed actions, estimated costs and **LAST ERROR**. A skipped action is not necessarily a malfunction.
3. To stop automatic posting, save Paused. **REMOVE WORKER** deletes that schedule, not posts already published, the agent runtime or its OAuth authorization.
4. **LIVE X WRITE GATE CLOSED** requires the platform operator to enable the service. Users can still save configuration/run permitted dry tests; no amount of repeated clicking opens this gate.



19. Telegram setup and identity

Choose the correct route first. To receive **Runcheck Agent** browser-order updates, use the shared bot supplied by RUNCHECK: sign in with your wallet at /full and create its private Telegram link. **You do not need to create an agent, a BotFather bot or a model API key.** Skip the dedicated-bot setup below and use “Link Runcheck Agent's shared Telegram bot”. Create **Telegram Assistant** only if you want your own separate AI agent with its own bot identity and model configuration.

Dedicated Telegram Assistant bot

This is your AI agent accessed through your own Telegram bot. You send it a private message and it replies in that same authorized chat using its configured model. For example, ask “What is funding on a perpetual market?” No second agent is needed. This template does not receive Runcheck Agent's browser-order updates and is not required to enable them. Creating it does not establish agent-to-agent communication, group moderation or automatic notifications.

1. Open the official [@BotFather](#), send /newbot, choose a bot name and username, and copy its token privately. Open the newly created bot (not BotFather or @myidbot) from the recipient's Telegram account and send /start to initialize the private chat. [Official bot setup instructions](#).
2. For a RUNCHECK-created Telegram Assistant, enter the BotFather token and **PRIVATE TELEGRAM CHAT ID** in the secure deployment form. For an external bot, enter the token and **AUTHORIZED CHAT ID** in its identity form. Both chat fields mean the recipient. RUNCHECK obtains **BOT ID** and **BOT USERNAME** from Telegram automatically; do not guess or enter them manually. External proof also requires the indicated Bot-to-Bot Communication Mode and the independent observer.
3. Use your positive private user/chat ID, not an @username, phone number or negative group ID. Obtain it from [@myidbot](#) in a private chat, or use the local export procedure below.
4. For an external bot, use **BIND TELEGRAM IDENTITY**, then check the verified identity. **UPDATE TELEGRAM IDENTITY** securely replaces that connection when allowed. This installs the credential for verification but does not deploy or take over your external bot runtime. For a factory agent, use **HAND OFF CREDENTIALS & DEPLOY**; there is no extra manual identity step.
5. For RUNCHECK agents, wait for the automatic model API check and try the chat. Only external bots need section 13's Telegram challenge: the observer must see the actual fresh reply. An arbitrary public t.me URL is

not proof of a private conversation.

6. Send a normal private message as the authorized user and inspect the reply. A different user's chat or a group is outside this configured scope. For a RUNCHECK-created bot, do not reuse a token already operated by another polling/webhook application. For external verification, keep your existing bot runtime running: RUNCHECK verifies its identity and reply without replacing its polling/webhook setup.

Which Telegram ID belongs in which field?

Field or destination	What to enter	Example / restriction
PRIVATE TELEGRAM CHAT ID / AUTHORIZED CHAT ID	The recipient's personal numeric Telegram user ID.	123456789: digits only, positive, without quotes, spaces, @ or +. Not a phone number. The recipient must first send /start to this exact bot.
BOT ID	The bot's numeric identity, obtained automatically from Telegram using its token.	Read-only identity, not the recipient's private chat ID.
BOT USERNAME	The bot identity obtained automatically from Telegram.	Not the recipient's personal username.
Operator alert group	A separately configured group destination for platform notifications.	Group IDs are negative, often starting with -100. They are not accepted by the agent's private-chat field.

The dedicated Telegram agent currently serves its configured private owner chat, not a group or channel. Adding the bot to a group or making it a group administrator does not replace this setup. Open your dedicated bot from the same personal account, press Start or send /start, then save its credentials and your private chat ID in RUNCHECK.

Use the dedicated bot after deployment

The commands below belong to RUNCHECK-created Telegram Assistant runtimes. An external bot keeps its own commands and behavior; verification does not add these features.

Send /help, /status, /markets or /chart BTC 1h, or write a normal message. To inspect a public Lighter portfolio, send /positions 0xYOUR_EVM_ADDRESS with a complete non-zero EVM address. This reads real mainnet account data; it never signs or submits an order. Repeated reads can reuse a snapshot for up to 15 seconds. If a venue is unavailable, the bot reports incomplete data rather than claiming the account is empty.

The dedicated bot has no wallet-link session: /positions without an address cannot infer your wallet. /link and /unlink are not advertised in its help and do not establish a connection. To share authenticated wallet context, use the separate central Runcheck Agent flow described below and the exact bot shown by that screen. Do not send its one-time link code to your dedicated bot.

The dedicated bot also does not receive the shared Runcheck Agent's wallet-linked execution receipts. Its normal role is to answer your messages and supported read-only commands, not monitor another agent and send unsolicited updates.

Find your ID with @myidbot

1. Open [@myidbot](#) in Telegram and press Start in a private conversation, not in your group.
2. Ask for your personal ID using the bot's commands/help; copy the positive numeric user ID it returns, not a group ID.
3. Paste only that number into **PRIVATE TELEGRAM CHAT ID** or **AUTHORIZED CHAT ID**. This ID-finding bot is a third-party utility, not your agent bot or BotFather; it does not need your bot token, model API key or wallet credentials.

Alternative: find your numeric user ID with a local export

1. In Telegram Desktop, use **Settings** → **Advanced** → **Export Telegram data**. Choose account information and JSON format; chat/media export is unnecessary. Complete Telegram's security confirmation if requested.
[Official export instructions](#).
2. Open the exported JSON locally, find `personal_information`, then copy only `user_id` into RUNCHECK's private-chat field. This is the field written by [Telegram Desktop's official exporter](#). Keep the export private; do not upload it to RUNCHECK or an ID-finding service.

Link Runcheck Agent's shared Telegram bot

RUNCHECK supplies this bot. The link connects your wallet's supported browser-order updates and optional AI chat to your private Telegram conversation. No agent creation, personal bot token or model API key is required. It neither creates a Telegram Assistant nor connects any personal agents to one another.

1. Open [Runcheck Agent](#), sign in with the wallet whose browser orders you want updates for and open **TELEGRAM NOTIFICATIONS**.
2. **CREATE PRIVATE LINK**, open the bot identified by the screen and send the one-use `/link CODE` privately if the deep link has not completed it. Never post the code in a group.
3. Return and **REFRESH LINK STATUS**. Confirm the bound Telegram identity. **Browser order updates** and **AI replies in Telegram** start enabled after this explicit link. For updates without conversation, keep **Browser order updates** on and turn **AI replies in Telegram** off. These preferences are independent.
4. If **AI replies in Telegram** is enabled, you may test a conversation or market question. For notification-only use, verify the linked identity and saved preferences without placing a trade. Any later trading action still requires browser review/signing; Telegram cannot submit a trade.
5. **UNLINK TELEGRAM** removes the wallet link. If access is denied, the operator must authorize your numeric tester ID; an expired code needs a new private link.

This shared platform connection and the dedicated Telegram Assistant are independent choices; neither requires the other. They are not the separate Telegram trading-bot application. Public launch announcements go to the operator-configured group through a third platform service, not your personal agent bot.

Notifications: who sends what to whom?

Message	Sender and recipient	Trigger / limit
A reply from your Telegram Assistant	Your dedicated bot replies to you in its configured private chat.	You send a question or supported command. This is conversation, not an automatic notification.
Runcheck Agent browser-order update	RUNCHECK's supplied bot sends the recorded update to the private chat linked to your wallet. No personal agent is involved.	Browser order updates must be enabled. A supported manual Lighter browser submission/reconciliation records an acknowledgement pending verification, verified position/protection checks or verified flat state for the selected market only. An acknowledgement is not a confirmed fill; that market being flat neither proves the whole account is flat nor that a take-profit triggered.
Public project announcement	The platform's announcement bot posts public project information to the operator-configured RUNCHECK group.	An eligible public agent profile or confirmed token launch enters the announcement service. It is separate from your private bot and does not reveal private requests or credentials.

For the shared connection, inspect **Browser order updates** and **AI replies in Telegram** after linking; new links enable both. Switch only AI replies off for notification-only use. Turning off updates removes undelivered order updates for that chat; unlinking removes both the connection and undelivered updates. These messages are tied

to supported manual Lighter browser submissions/reconciliation, not continuous monitoring. No price alerts, automatic take-profit/stop-loss-hit alerts, arbitrary reminders, agent-to-agent messages or Autotrader notifications are configured by either preference.

20. Lifecycle and troubleshooting

Open USER → **MY AGENTS** → the exact agent's **RUNTIME** view for its deployment state and lifecycle links. If credential, service or retirement controls open setup, they keep the same agent in AI + deployment (stage=model). Use **ACTIVITY** for that agent's recorded history; **CONTROL ACTIVITY** in USER lists owner activity across agents. Archived runtime/services views are recovery-only: use the retained proof, vault, Autotrader and activity views rather than trying to reconfigure an archived deployment.

End a test cleanly

1. Save transaction hashes, agent ID and private execution report. Check wallet, vault and Lighter balances independently.
2. Pause/remove X schedules. Stop real Autotrader signing, revoke its venue key and resolve remaining orders/positions. Removing a browser signer alone is insufficient.
3. Recover vault assets to the owner, revoke test sessions and pause the vault if appropriate. Keep enough wallet ETH for recovery transactions.
4. Read/tick the runtime retirement confirmation and **RETIRE RUNTIME** (BYOK) or the Managed AI control in section 12. Use **RETRY RETIREMENT CLEANUP** if required. Check that cleanup, not merely the request, finishes.
5. Revoke unused Managed X consent and underlying connector access as appropriate; unlink Telegram. Retiring one runtime does not erase already-published posts or on-chain tokens.
6. Archive only an eligible clean draft/retired agent, using the exact displayed confirmation phrase. A current proof or active launch can block archive. Archived-vault recovery remains a separate owner operation.

Symptom	First recovery action	Who resolves a persistent blocker
Buttons absent / wrong owner	Check selected agent and full owner address; sign in again on chain 4663.	User; operator for a reproducible identity error.
Name / public address already reserved	Read the error by the registration fields and choose a different unique name. A saved or archived registration may retain its address.	User; operator only if reservation ownership is inconsistent.
Saved draft but next screen did not load	Use REFRESH SAVED AGENT . The draft was saved; do not register another copy or sign repeatedly.	User; operator if status reads keep failing.
Bot cannot access private chat	Check the recipient ID, open the exact bot associated with this token and send /start from that recipient account. Then retry the credential handoff.	Recipient; operator only if the same valid setup remains rejected.
Deployment queued/failed	Inspect the existing job, model access/credit and connector identity; retry that deployment.	User for keys; operator for worker/provider outages.
Paid but no runtime	Confirm the original transfer and entitlement, then provision; do not pay again.	Operator if receipt and ledger disagree.
Proof expired/revision changed	Factory: reopen/retry the automatic model API check. External: start a fresh challenge for the current identity/revision.	Operator for factory service errors; runtime developer for an incompatible external endpoint.

Symptom	First recovery action	Who resolves a persistent blocker
External verification busy	Keep the existing attempt and retry after the displayed cooldown; do not repeatedly create challenges.	Operator if capacity/observer failure persists; agent developer for incompatible responses.
Private GitHub review awaits access	Connect/revalidate the App for the exact repository, then use VERIFY GITHUB APP + QUEUE . For the manual invitation/read-token fallback, ask the operator to verify restricted API access and approve the request. Never paste tokens into the brief.	Repository owner; RUNCHECK operator for App configuration or the manual fallback.
Repository queued / changed / stale	Read the commit and last-check time. Refresh reads saved status; it does not bypass the shared scan budget. Check webhook setup or wait for the scheduled check.	Operator for a stopped worker; repository owner for webhook permissions.
Dependency check unavailable / partial	Read the advisory status and manifest coverage. Ask the developer to supply supported exact versions or retry after provider recovery. A static no-match result does not clear uninspected dependencies.	Project developer; operator for persistent lookup failures.
Lighter account missing / zero margin	Check Robinhood deployment and numeric account index; complete/confirm venue deposit.	User/venue support for deposit issues.
Autotrader waits	Read signal, flat-account state, interval, collateral, budget and expiry.	User for policy; operator for missing worker heartbeat.
Browser signer fails	Reload the current site, reconnect owner/account; preserve pending hashes and report the error.	Operator for repeatable CSP/WASM/connection failures; do not disable browser security.
Unknown trade/swap	Use saved-intent reconciliation and check venue/chain before another submission.	Operator when authoritative state conflicts.
Pons busy / 429 / 503	Wait at least five seconds, retry once, prepare fresh before signing.	Operator if sustained; no burst-refresh loop.
Token operation disabled	Check quote asset, current recipient, escrow, vesting and graduation/anti-snipe phase.	External Pons operator for operator-only conversion/phase blockers.
X intent locked	Check status/evidence, not another publish.	Operator/provider if ambiguous result persists.
Telegram fails proof / link	Check private ID, correct bot, Start, fresh code/challenge and bot-to-bot setting.	Operator for observer/allowlist; user for own bot configuration.
Retirement still pending	Stop live activities first, retry offered cleanup and retain records.	Operator for unresolved cleanup; do not create a duplicate runtime to bypass it.

Support report: page URL, UTC time, agent ID, selected chain/account, exact button/error and public transaction hash when present. Redact account exports, private reports and all credentials from screenshots. A user test passes only when its expected result is observed, not because a button was clickable.

21. Feature completion checklist

Use this as a progressive exploration checklist, not a declaration that transactions have already been tested. Every public feature in the product catalog has a procedure below. Admin, infrastructure and private credentials are deliberately excluded.

Public feature	Procedure	Evidence to record
Wallet identity	10	Correct address/chain and owner-bound signed session.
BYOK factory	11	Chosen template and ready runtime revision.
Runtime isolation	11	Two agent cards retain their own configuration/credential references.
Agent vault	16	Creation, deposit/recovery and session revocation receipts.
Agent lifecycle	20	Retirement cleanup; eligible archive and retained recovery.
Managed AI factory	12	Quote → one payment → entitlement → ready runtime; credit/top-up ledger.
Runcheck Agent	4, 14	18 grouped research prompts, source links, asset-matched 1h/4h/1d chart, voice, Markets, owned-agent Vaults & auto and optional Telegram; separate trading approvals.
Telegram agent	19	Dedicated private bot reply with correct owner identity.
Social X agent	18	Verified post/reply/quote and reviewed paused/enabled schedule.
Managed X	12	Account-bound paid period, renewal and consent revoke/re-enable.
Paper Autotrader	13, 15	External structured paper proof; not required to create/activate a factory trader and not real execution.
Real Lighter Autotrader	15	Advice/custom policy, signed delegation, actual order/protection, report and stop.
External agent proof	13	Registered endpoint/identity and proof-only certificate.
Custom agent requests	13	Submit the brief, inspect REQUESTED → APPROVED → PUBLISHED, then open the delivered agent/token/certificate from the private request detail.
Agent code audit	13	Submit any agent's repository audit; inspect commit, limited coverage and findings; verify privacy, monitoring warnings and withdrawal.
Capability challenges	13	Fresh manual challenge for external agents; automatic model response for factory agents.
Public certificates	13	Factory model check or external capability, correct revision/expiry, accessible signed out.
X proof	13	Exact bound author and fresh public-post evidence.
Telegram proof	13, 19	Independent observer result, not a self-reported reply.
Launch gate	13, 17	Current factory model check and all launch prerequisites; stale/absent proof remains blocked.
Pons wallet launch	17	Exact V2 review, owner signature and verified token.

Public feature	Procedure	Evidence to record
Launch reconciliation	17, 20	Pending/reloaded launch reconciles to one final result.
Agent directory	10, 13	Search/filter, publish/unpublish and public profile.
Pons market	10, 17	Browse + buy/sell, sweep/claims/vest, fee rights, graduation/pool when eligible.
Verified agent-token link	10, 17	Navigate token → agent → current certificate.
Lighter browser execution	14	Local enrollment, exact entry/close, protection and reconciliation.
0x spot routing	14	Indicative/firm quote, exact approval, swap and confirmed minimum receive.

For each test, record date/time, agent/token/account identifier, expected versus observed result and a receipt or redacted screenshot. Mark operations requiring a later vesting/graduation phase as **not yet eligible**, not “passed”. Stop and investigate a failed financial reconciliation before continuing that account's tests.

22. Practical use cases

RUNCHECK's central use case is a wallet-owned AI-agent launchpad on Robinhood Chain: create a supported agent, obtain its automatic model-response certificate, operate its features and optionally launch a linked Pons V2 token. External developers can instead prove one supported capability of an agent they already host. Neither path implies that all products, identities or trading permissions are enabled at once.

Goal / role	Path to follow	You need	Completion looks like
Explore as a visitor	Platform → Agents / Tokens → profile / certificate	No wallet for public browsing	You can identify the product, proof scope and token link.
Create a personal AI agent	Agent & Token Launchpad → Personal Companion → BYOK or Managed AI	Owner wallet; provider access or paid credits	Your own AI web chat and current model-response certificate; no shared platform trading/notification panels.
Launch an agent token as creator	Ready factory agent → Launch Gate → token form → wallet	ETH for gas/launch; chosen quote asset for initial buy	One reconciled Pons V2 launch linked to your agent.
Verify an existing external agent	Verify Agent → external verification → supported proof → endpoint or identity	Owner session; compatible external service or X/bot evidence	Public capability certificate; no factory hosting or token eligibility granted.
Operate a perpetual strategy	USER → MY AGENTS → AUTOTRADER advisor → review → delegation	Ready trader, exclusive funded Lighter account, enrolled key	Owner-authorized worker with visible decisions, orders and stop controls.
Publish project updates	Social X → connect → publishing / engagement	Own X account, AI access, service/consent where required	Reviewed posts and a controllable schedule tied to the correct author.
Use the shared research/trading workspace	RUNCHECK menu → Runcheck Agent at /full	No agent creation or own model key; sign in and authorize operations when needed	Shared research chat and separately authorized manual browser trading.
Receive browser-order updates in Telegram	Runcheck Agent → TELEGRAM NOTIFICATIONS → private link	Wallet sign-in and your private chat with RUNCHECK's supplied bot; no own BotFather bot or agent	Supported manual Lighter order updates; AI replies can be switched off independently.
Create your own private chat agent	Agent & Token Launchpad → Telegram Assistant	Your BotFather token, authorized private chat and model setup	Your dedicated AI bot answers your messages; it is not the shared notification service.
Operate capital or tokens	USER → MY AGENTS → VAULT, or Tokens → V2 operations	Correct owner/beneficiary wallet and relevant balance/phase	Confirmed deposit/recovery, swap or eligible claim with a receipt.

A. A creator launches a Personal Companion and its token

1. Browse the public site, connect the intended owner wallet and select **CREATE AN AGENT → PERSONAL COMPANION**. Name it "Market Notes Demo"; use the description/character examples in section 11.

2. Save the draft. Choose BYOK and a model available to your provider account, or purchase and confirm the displayed Managed AI pack. Deploy once.
3. Wait for the automatic model API check. Inspect its certificate; do not enter **VERIFY MY AGENT** for this factory agent. Open its own AI web chat. Runcheck Agent's shared market, manual-trading and Telegram notification panels remain a separate platform tool, usable without creating this agent.
4. Publish the profile if wanted. For the optional token, continue to Launch Gate, select ETH if you want an ETH pair, upload public artwork, review fees/initial buy and sign the exact launch.
5. After reconciliation, follow token → agent → certificate. Test only the eligible operations in section 17. A buy or fee claim is a new wallet operation, not part of agent creation.

Result: a configured factory agent, its model-response proof and, if you signed the launch, a linked tradable token. The token does not itself start a strategy, fund a vault or grant holders access to the owner's private agent controls.

B. An external developer verifies a custom assistant

1. Keep the agent hosted on your own service. Choose **VERIFY AN EXTERNAL AGENT**, then the response or paper-proposal test matching what that endpoint can demonstrate. This is not the route for an approved custom RUNCHECK delivery.
2. Register its name and actual public HTTPS challenge URL. A general support/research assistant can use Companion Response if it implements that protocol; it need not have been built from RUNCHECK's Personal Companion template.
3. If you only have a chat website, ordinary OpenAI-compatible API or an authenticated endpoint, ask that runtime's developer to supply the supported public challenge/health interface. Do not put API secrets in the URL.
4. Run the fresh challenge and inspect the capability certificate. For an external X identity or Telegram bot, use the identity-proof route instead of the URL route.

Result: proof of one supported behavior, not migration into RUNCHECK, AI credit provisioning or access to the factory-only token launch. "Custom" does not mean RUNCHECK can test every arbitrary feature.

C. An owner turns a recommended setup into real Autotrading

1. Create a RUNCHECK Autotrader and wait for its automatic model check. A token launch is optional; no manual paper challenge is required.
2. Fund and reserve an exclusive account on Robinhood Lighter. Enter its numeric index in the selected agent's **AUTOTRADER** view from **MY AGENTS**, not the vault address.
3. Ask the advisor for a strategy using your own collateral and limits. Read the rationale/simulation and apply the returned setup only when applicable. Application fills fields; it does not trade.
4. Register the dedicated key, review the final policy, then sign **SIGN DELEGATION & ACTIVATE REAL TRADING**. Inspect decisions, venue-confirmed fills and protection. If no signal qualifies, waiting is an expected result.
5. Test pause and inspect the unchanged existing position. When finished, stop/destroy the worker key, revoke it at the venue and resolve remaining positions/orders. Download the private report.

Result: real autonomous execution under an explicit delegation. A browser chat suggestion, paper certificate, disconnected wallet or closed tab is not the control that starts or stops it. See section 15 for precise boundaries.

D. A project owner publishes updates and adds private Telegram

1. Create a Social agent with purpose, language and approved sources. Authorize the exact X account; complete BYOK and any selected Managed X subscription/consent separately.
2. Publish one reviewed post and open the returned evidence. Configure the engagement worker paused, run the labelled dry test and inspect its decision/cost.

3. Enable and save the schedule only when ready for real public posts. Monitor counters and pause/remove the worker when finished. Revoking Managed X consent and revoking X OAuth are different actions.
4. For shared market conversation or browser-order updates, open Runcheck Agent and link its supplied Telegram bot. You need neither a personal agent nor your own BotFather bot. For updates only, turn off **AI replies in Telegram** and leave **Browser order updates** on. Create Telegram Assistant only if you want your own separate AI bot; never reuse a bot token operated by another application.

Result: account-bound communication you can control. Managed AI credits, Managed X, a dedicated Telegram bot and the shared Runcheck Agent link are distinct choices, not one bundled subscription.

E. A token holder or vault owner operates existing assets

1. As a token holder, open Tokens, inspect contract/quote asset/proof, connect your own wallet and review BUY or SELL. You do not need to create an agent to trade a supported token.
2. As a creator/beneficiary, check escrow, vesting and current fee rights before using claims or fee controls. A buyer is not automatically the creator-fee recipient. Wait for eligible graduation/vesting phases instead of trying every button immediately.
3. As an agent owner, open that agent's **VAULT** view from **MY AGENTS**, review creation if needed and sign a deposit. Verify the vault balance, then review withdrawal back to the owner. Vault creation is a separate operation, not an automatic consequence of launching a token.
4. Review session permissions only if you need scoped on-chain access. Vault sessions do not supply Lighter collateral or activate the Autotrader. Preserve receipts and check final balances before moving to another asset.

Result: confirmed operations against the exact token or agent vault, using the correct role. See sections 16–17 for each action and its recovery path.

23. Everyday reference and FAQ

Common questions

Question	Answer
Must every agent have a token?	No. In the current product, agents can operate without a token. The RUNCHECK token-launch path does require an eligible current factory agent.
Do native and custom RUNCHECK agents need manual verification?	No. Native agents and approved custom deliveries receive scoped certification automatically when the required current checks and delivery evidence agree. Inspect saved proofs in USER, not external verification. Missing, failed or stale evidence must still be resolved; a platform badge is not a GitHub code audit.
Does “custom” mean only Companion or Autotrader?	Those URL-route labels select supported response/paper tests, not how the external agent was built. X and Telegram have separate identity tests. Unsupported abilities are not certified.
Is a GitHub report a guarantee that an agent is genuine or safe?	No. It records an exact commit, bounded static checks and explicit coverage. A runtime capability certificate is separate; neither proves the live runtime matches that repository.
Does RUNCHECK send my private code to an AI model or advisory provider?	Static is offline and removes transient sandbox source on exit. Separately consented OSV sends up to 100 package/ecosystem/version tuples, not source; names may disclose private dependencies. Intelligence requires fresh consent v2 for its commit/profile-bound source transfer: up to 8 batches, 64 files/batch, 64 KiB/file, 128 KiB/batch and 1 MiB overall, sharing 15 seconds and 8,192 output tokens. Redaction cannot guarantee removal of every confidential detail. Review provider/profile and processing terms; revocation cannot undo prior processing.
Will every push immediately receive a new audit?	No. A configured authenticated webhook invalidates relevant old evidence and queues a check. Global pacing, repository cadence and provider availability still apply; every intermediate commit is not guaranteed a separate report.
Is Runcheck Agent the agent I just created?	No. Runcheck Agent at /full is a shared platform tool that needs no agent creation. Personal Companion is your own AI web-chat agent; its template does not include the shared manual-trading or Telegram update panels.
Must I create Telegram Assistant to receive order updates?	No. Link your wallet's private chat to the bot supplied by RUNCHECK inside Runcheck Agent. No BotFather token or model key is needed. Keep Browser order updates on and switch AI replies in Telegram off if you want updates only.
Will Telegram tell me every time a take-profit or stop-loss triggers?	No. These are recorded updates from supported manual Lighter browser submissions/reconciliation, not continuous position monitoring. They do not promise automatic TP/SL-hit, price or Autotrader alerts.
Does a token provide AI credits or revenue to every holder?	No such entitlement follows automatically. AI packs are purchased separately; creator fees/vesting follow the token's displayed on-chain rights.
How many AI packs can I buy?	The current catalog offers one standard pack type, one per checkout, with repeatable confirmed top-ups. Current price/credits/daily limit are in Docs → AI credit pack. More credit does not raise that daily limit.
Can I use my own AI provider?	Yes, through a supported BYOK provider/model or compatible endpoint. Model access and API billing belong to that provider account. It is separate from RUNCHECK Managed AI.
Why USDG if my token pair is ETH?	ETH pairing governs that token's quote asset. USDG pays the displayed hosted services and is the supported same-chain Lighter collateral; those are separate balances/purposes.

Question	Answer
Does signing in or launching start Autotrading?	No. Only the reviewed Autotrader activation delegation starts its worker. Key enrollment, applying advice and token launch are not that final activation.
Does closing the browser stop everything?	No. Delegated trading and enabled publishing workers can continue. Pause/stop/revoke them using their controls; resolve remaining positions separately.
Are vaults, Lighter accounts and token pools the same?	No. They hold different capital under different rules. Depositing into one does not fund the others.
Can Telegram execute my trades?	Runcheck Agent's shared Telegram bot routes approval back to the browser. The separate Telegram trading-bot product is not part of this launchpad guide or a dependency.
Why is a button unavailable?	Read the named requirement: owner/session, model check, service, credits, identity, balance, pending transaction or on-chain phase. A later-phase operation is not a missing setup step.

Short glossary

Term	Meaning in this dApp
Agent / template	Your registered software identity / the supported starting configuration used to create it.
Runtime / revision	The deployed process / the specific configuration being checked. A draft is not a running process.
Factory agent / external agent	Deployed through RUNCHECK / hosted elsewhere and registered for supported proof.
Model API check / capability certificate	Evidence of a fresh model response / evidence of the precise named test, scoped to identity, revision and validity.
Native launch / association / Certified AI	A finalized token launch submitted through RUNCHECK / an existing Pons V2 token connected later / a separate current token-bound capability proof. None implies the others.
Launch Gate / reconciliation	Prerequisites before a token signature / checking a submitted action against its authoritative final outcome.
BYOK / Managed AI / Managed X	Your provider key / RUNCHECK-managed inference credits / separate account-bound X publishing subscription.
Quote asset / gas	Asset spent/received in the token pair / ETH network transaction fees.
Collateral / notional / turnover	Margin backing a position / gross position size / cumulative entry activity counted by policy.
Basis point (bps)	One hundredth of one percent. 100 bps = 1%; units differ from an absolute price.
Delegation / session / allowance	Scoped worker authorization / scoped vault access / token-spender permission. None is equivalent to login.
Escrow / vesting / graduation	Credited claimable balances / scheduled token release / eligible transition from curve trading to its pool.

Returning each day, and finishing safely

1. Confirm site URL, header owner/network and the selected agent/account. Refresh health, credits/entitlements and outstanding operations before acting.
2. For active strategies or schedules, inspect the latest decision, actual venue/post evidence, limits, expiry and errors. A service ONLINE badge is not proof of a successful order/post.

3. Reconcile pending transactions before another attempt. Keep only the identifiers needed for support, never private keys or bot/link codes.
4. When ending an experiment, pause new activity, resolve positions/orders, stop and revoke the relevant keys/consents, recover intended capital, then retire/archive eligible agents. Disconnecting alone is not cleanup.
5. Use section 21 to mark what you actually observed as completed, blocked or not yet eligible. Ask the operator about service/identity enablement; use section 20 for an actionable support report.