

AutoTrade

Market intelligence.

Decision control.

Coordinated execution.

An integrated platform for analysis, signal control and trade execution in
MetaTrader 5

PRODUCT & TECHNOLOGY BOOK / 2.0



01 / THE PRODUCT

What is AutoTrade, and what does it do?

AutoTrade is ARG’s platform for market analysis, signal generation and control, and coordinated trade execution in MetaTrader 5. Its web application, backend engines, desktop client and Expert Advisor connect the entire workflow: from market data and market-state interpretation to account policy, order submission and outcome reporting.

- 01

One shared market picture

Price, structure, regime and multi-timeframe analysis share one analytical workspace.
- 02

Decisions with context and policy

Independent engines produce signals; geometry, quality and account rules determine the executable path.
- 03

Execution you can trace

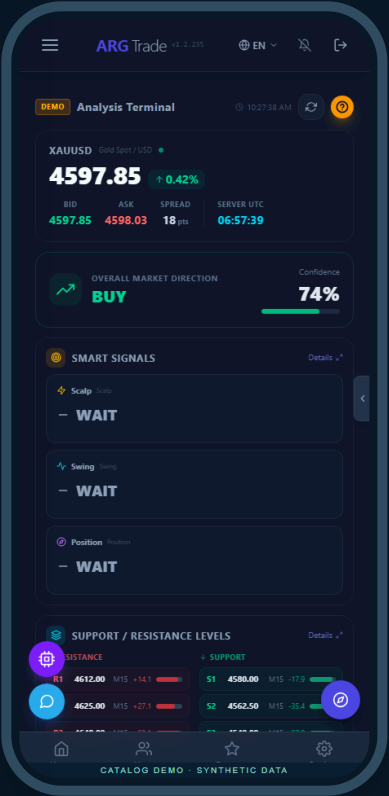
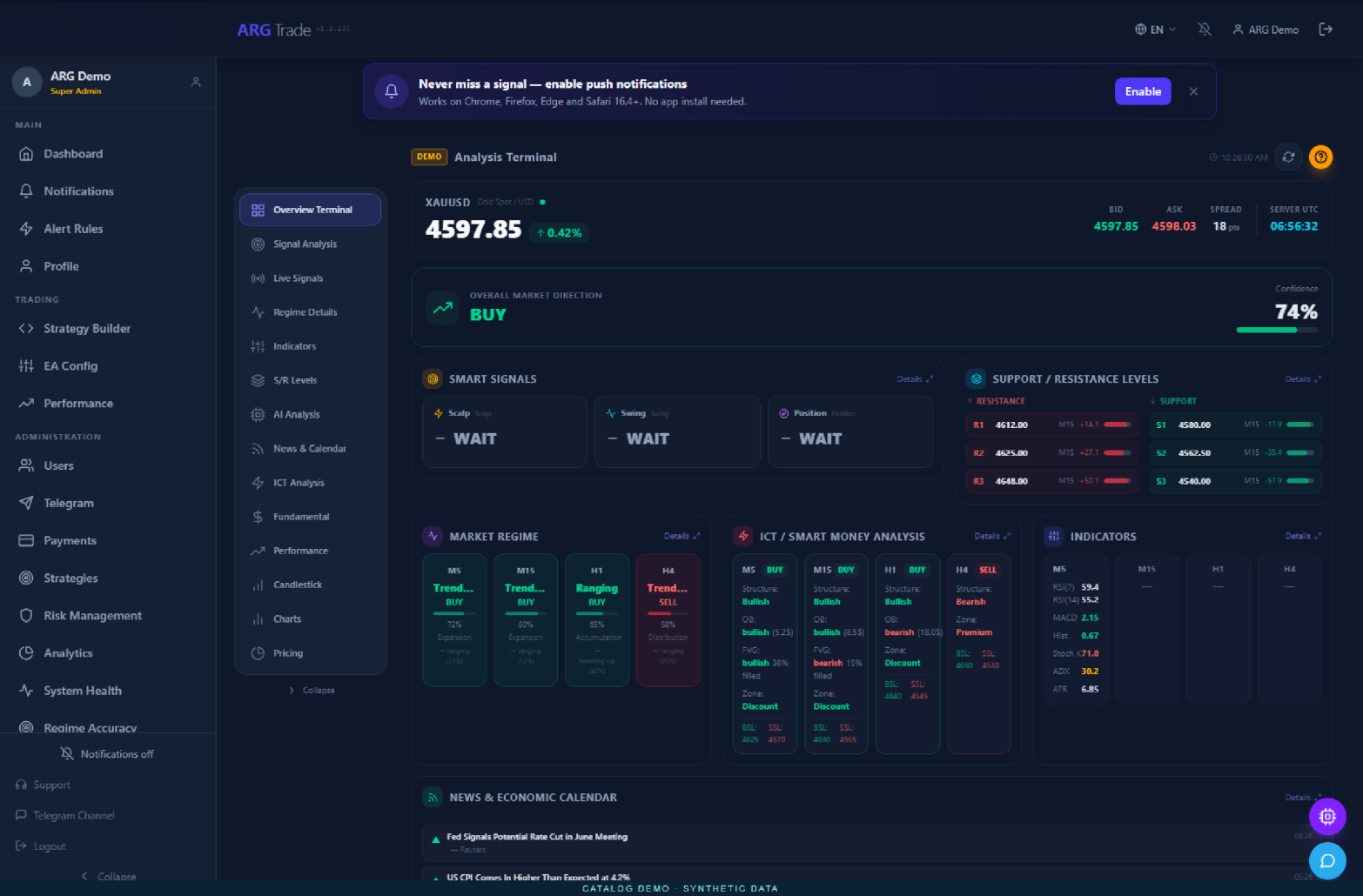
Signal and account identity, durable delivery and EA reports let operators inspect the path from decision to outcome.



02 / PRODUCT EXPERIENCE

The workspace for analysis and trading operations

From a market overview to detailed inspection: web and mobile views of the same product.



Actual application interface, captured locally with synthetic data. Values and directions shown are not real account results.

03 / PEOPLE & PURPOSE

Who is it for, and what work does it support?

Three complementary roles with different needs across one trading workflow.

01

Understand market conditions

Analyst & trader

Reviews the analysis terminal, ICT, support/resistance, indicators and regimes, then examines a signal's horizon and context before following its outcome.

02

Control accounts and execution

Trading operations lead

Manages access, EA configuration, account risk, terminal readiness and signal delivery, investigating why a signal waited or execution was rejected.

03

Develop and evaluate logic

Strategy & engineering team

Authors Builder rules, inspects the associated engine and policy, and evaluates replay, shadow and actual execution evidence separately.

04 / PRODUCT VALUE

Connect information, decisions and execution

The product's value lies in a connected workflow and explicit responsibility.

01

One shared market picture

Price, structure, regime and multi-timeframe analysis share one analytical workspace.

02

Decisions with context and policy

Independent engines produce signals; geometry, quality and account rules determine the executable path.

03

Execution you can trace

Signal and account identity, durable delivery and EA reports let operators inspect the path from decision to outcome.

Example workflow: an analyst reviews market context; an engine creates a candidate; path and account policies determine delivery eligibility; the EA checks the order against actual symbol constraints and reports the outcome.

05 / END-TO-END WORKFLOW

One complete workflow. Six explicit stages.

01

Ingest & validate

Quotes and candles are checked for clock contract, provenance and freshness.

02

Analyze & contextualize

The shared hub feeds market structure, regimes, indicators and decision engines.

03

Generate & evaluate

Comp and StrategyEngine maintain separate state and apply their own candidate policies.

04

Control & allocate

Geometry and account eligibility are checked; delivery retains identity and expiry.

05

Execute inside MetaTrader

The Bridge writes the file contract; the EA checks volume and submits orders to the broker.

06

Observe & evaluate

Delivery state, execution events and outcomes return to web and evaluation surfaces.

06 / CAPABILITY LANDSCAPE

Product breadth, defined precisely

These figures describe structure and interface scope, not speed or investment performance.

03

Analytical horizons

Scalp / Swing / Position

06

Feed-guard timeframes

M1 / M5 / M15 / H1 / H4 / D1

304

Variable-registry entries

Builder snapshot · b90f26a

61

Web page components

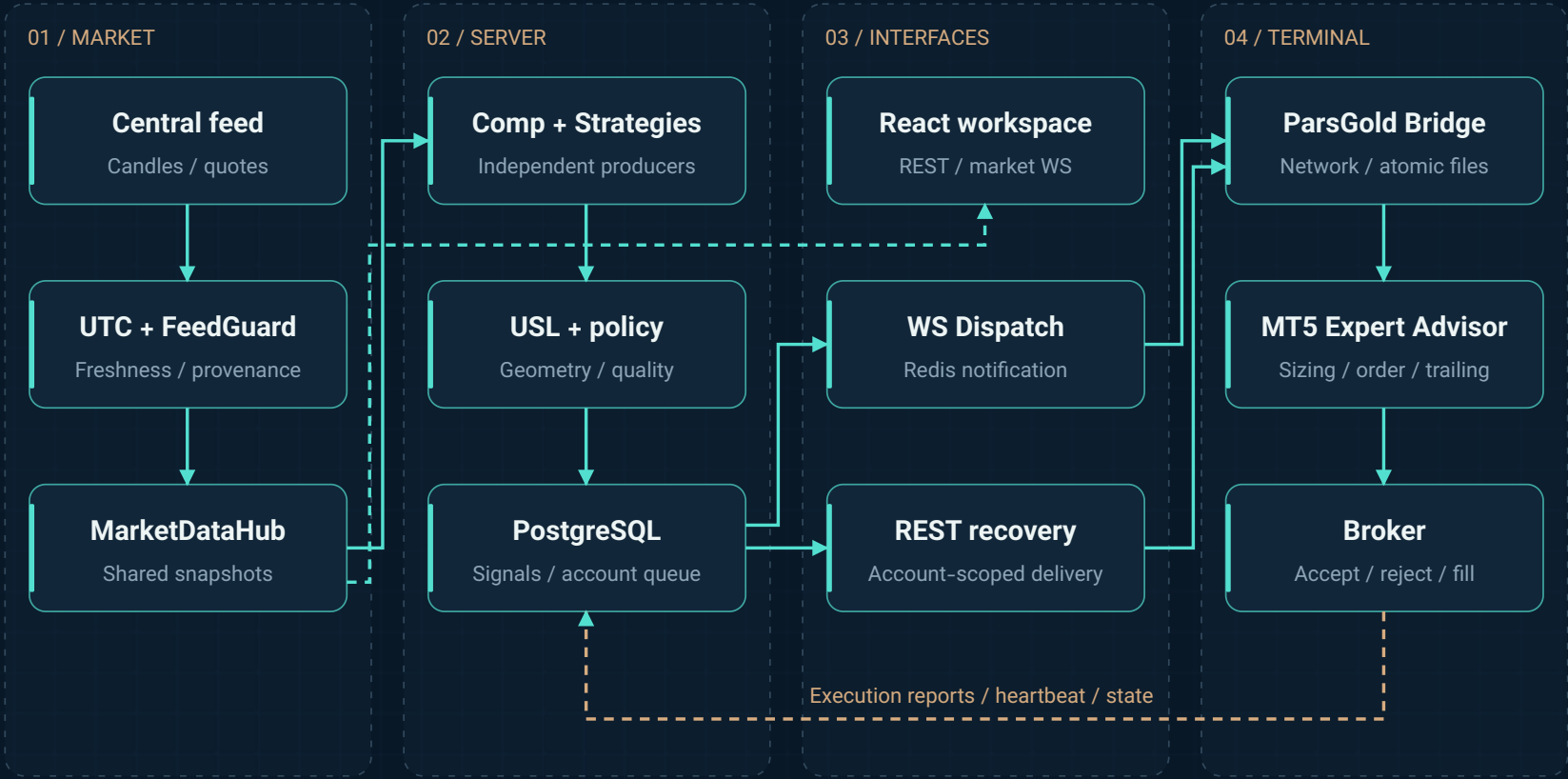
86 concrete routes

A web application for analysis and administration; backend services for data, signal engines and policy; a Bridge for terminal communication; and an EA for volume, orders and positions. Enabled strategies and features depend on configuration and installed versions.

The architecture from market feed to broker

THE AUTOTRADE SYSTEM

Analytical decisions, account delivery and broker execution have distinct owners.



Solid: decision & delivery Dashed: observation & feedback Source topology, not a deployment diagram

READING MAP

Start with the product. Then explore mechanisms and evidence.

Experience & data

10–15

Frontend and role-based experience / Identity, access and licensing / Market hub and a consistent clock / Feed validity and freshness gate / Layered market analysis / Market regimes and directional context

[Open chapter →](#)

Delivery, EA & execution

26–32

Durable per-account delivery / Protocols and communication contracts / ParsGold desktop client / The Expert Advisor inside MetaTrader / Position handling and trailing / Expert Advisor family and companion tools / Updates and activation evidence

[Open chapter →](#)

Evaluation & visual index

46

Evaluation criteria, 61 web pages, interaction details and 14 desktop views

[Visual index → 47](#)

Decision & strategy

16–25

Comprehensive engine and three horizons / V2 reversal lifecycle / Momentum and wave policy / Market-state routing / Unified Signal Layer / Strategy library / Visual Strategy Builder / Execution of authored rules / Structural stop and target construction / Risk controls across layers

[Open chapter →](#)

Intelligence, evaluation & operations

33–45

Entry-quality ML: observation mode / Quality, exhaustion and policy experiments / Trading hours and temporal learning / Daily policy and profit retention / AI services and analytical assistant / News, calendar and Telegram services / Rule-based notification engine / Backtesting, replay and evaluation / Operations and observability / Deployment and release lifecycle / Capability status and evidence / Additional Expert Advisor paths / Desktop research and modelling tools

[Open chapter →](#)

Routes & technical evidence

69

86 concrete routes, API modules and pinned source references

[Source evidence → 75](#)

01 / FRONTEND AND ROLE-BASED EXPERIENCE

One workspace. Multiple roles.

The React 19 web application uses Vite, lazy route loading and multilingual interfaces to combine analysis and administration.



Actual application interface, captured locally with synthetic data.

01 Interface structure

Public, authentication, analytical, operator and super-administration pages use distinct routes. This edition maps 61 distinct page components and 86 concrete routes in App.jsx.

02 Presentation tools

Price charts use lightweight-charts; visual editing uses React Flow; i18next handles translation and Framer Motion supports transitions. Seven locale files are present.

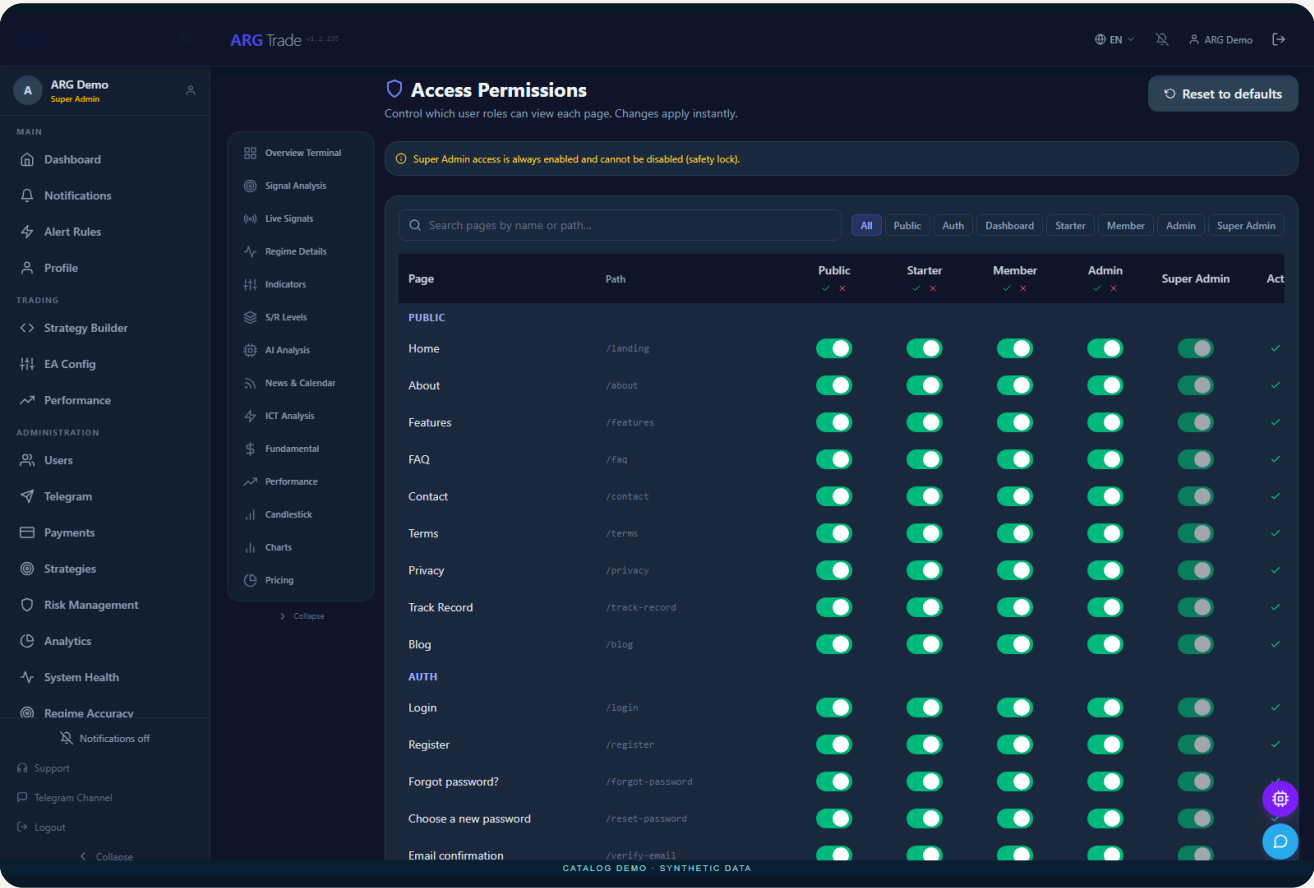
03 Data refresh

Market data uses REST and WebSocket. Concurrent GET requests are coalesced, with short caches and sessionStorage reducing repeated requests. UI visibility and server authorization remain distinct layers.

01 / IDENTITY, ACCESS AND LICENSING

Who can access what?

Login, roles, subscription entitlement and trading-account permission form a related chain with independent checks.



Actual application interface, captured locally with synthetic data.

01 Authentication

AuthContext validates the token through the profile endpoint and schedules refresh. A cached browser profile alone does not grant access. Web roles are public, starter, member, admin and super_admin.

02 Web access policy

AccessPolicyContext retrieves server-authoritative access and preview-time accounting. Page permissions merge defaults with server overrides; a route guard does not replace API authorization.

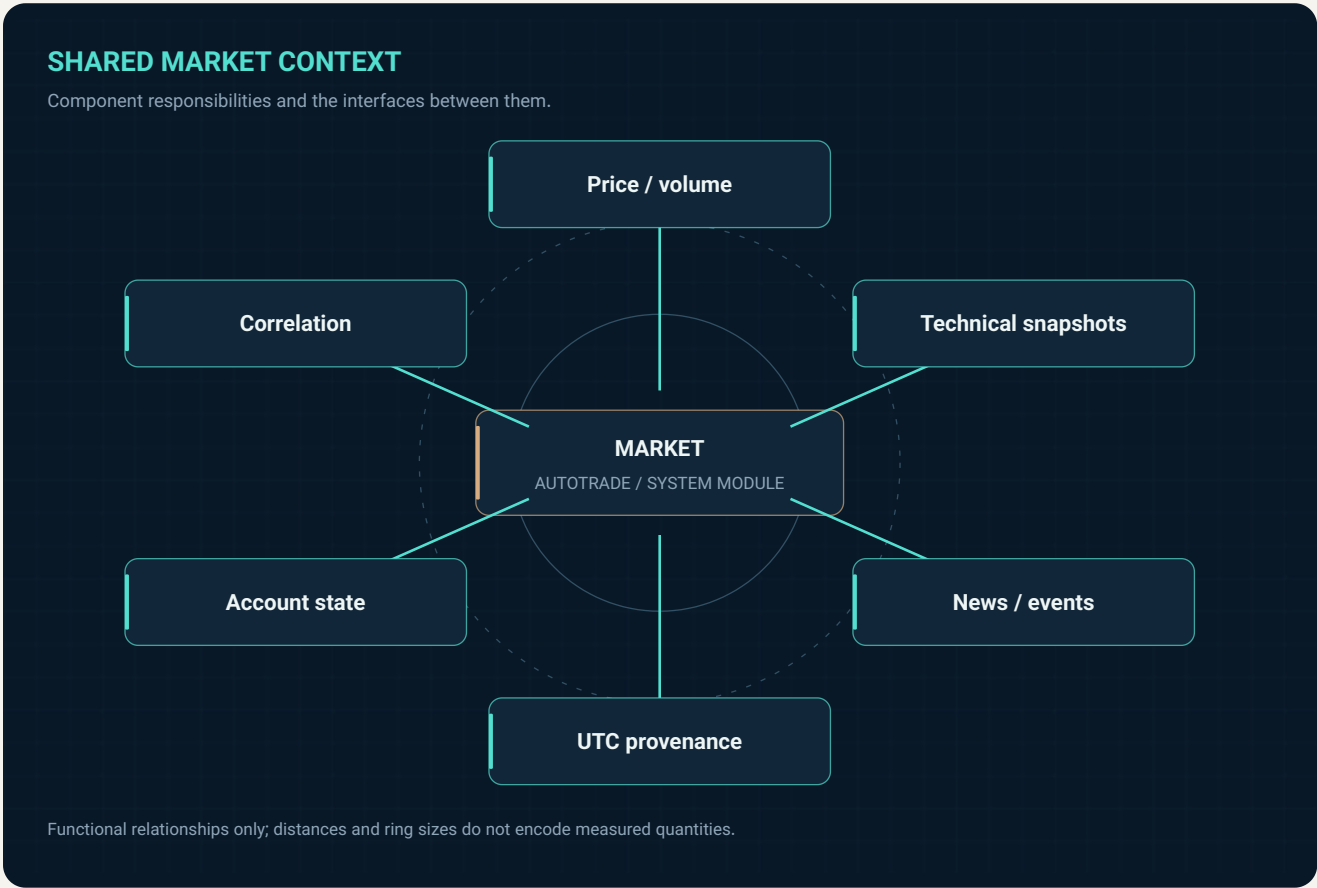
03 License service

ArgLicenseServer contains identity, licensing, EA configuration, subscription and payment functions. License validity, account ownership and terminal readiness are independent of the visible page.

01 / MARKET HUB AND A CONSISTENT CLOCK

A common context for every analysis

MarketDataHub gathers prices, candles, indicators, regimes and market context for analytical consumers.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Shared market snapshot

Typed snapshots organize price, volume, technical, correlation, account and news information. Consumers share this data layer to reduce inconsistent market views.

02 Candle time versus receipt time

The opening time of an H4 or D1 bar is not a transport-liveness measure. Receipt time, central-source provenance and the UTC contract are tracked separately.

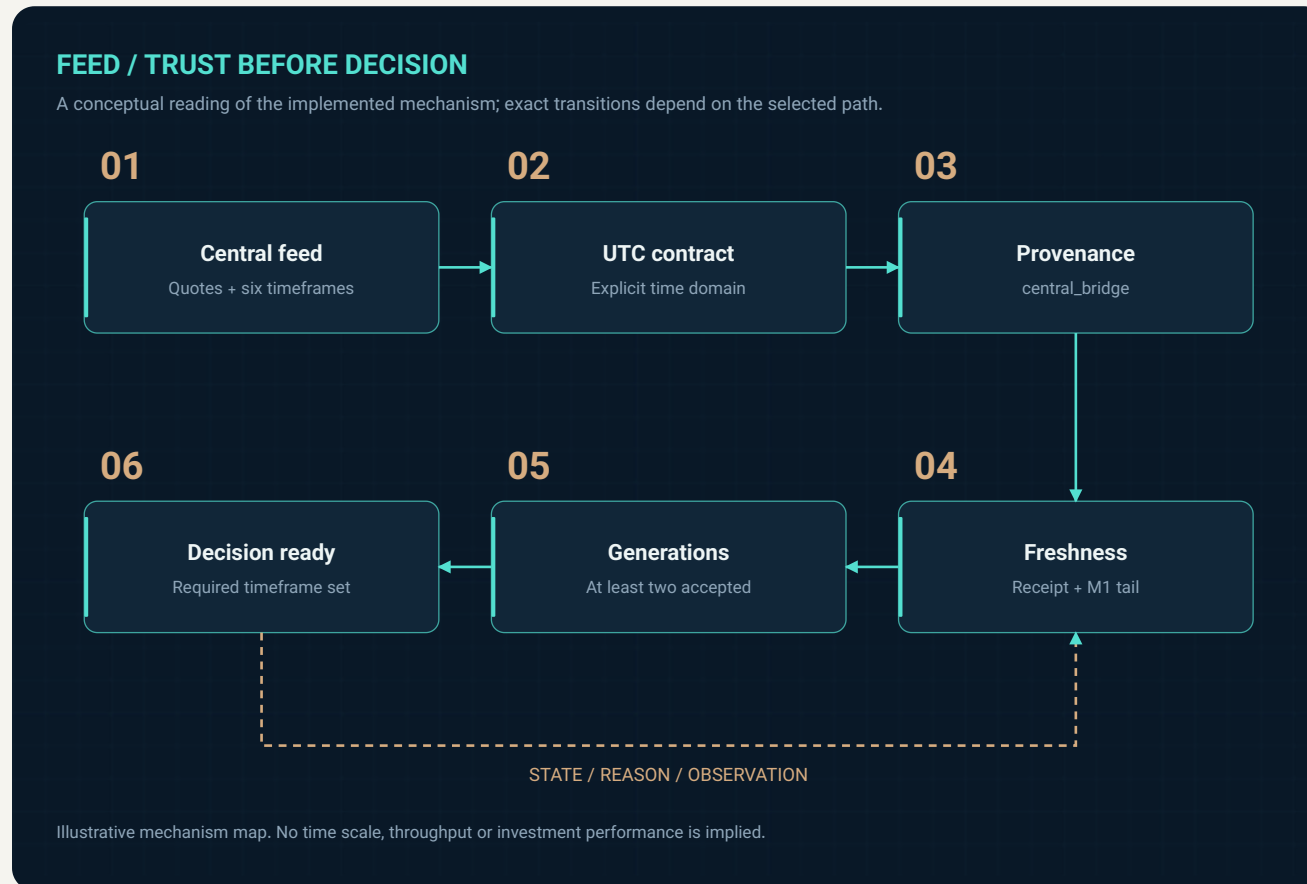
03 Restart continuity

Restored history must carry valid epoch and provenance. A fresh displayed quote alone does not prove that every timeframe required for trading decisions is ready.

01 / FEED VALIDITY AND FRESHNESS GATE

When is the feed ready for decisions?

Before tradable signal paths proceed, the complete required data set is checked.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Six timeframes

M1, M5, M15, H1, H4 and D1 are checked. The reference code uses a 180-second receipt-age limit and a 600-second M1-tail limit. These are source thresholds, not a service-latency guarantee.

02 Provenance and depth

The guard requires central_bridge, UTC, explicit-domain-v3, utc-v1 and at least two accepted generations. Missing, shallow, future-dated or stale data can block the path.

03 Operational effect

The guard reads the validated cache generation directly instead of a fallback that may return old data. Failure reasons are separated for diagnosis.

01 / LAYERED MARKET ANALYSIS

Read the market from multiple angles

Trend, levels, liquidity structure, indicators and market context provide traceable decision inputs.



Actual application interface, captured locally with synthetic data.

01 Trend and levels

trend_calculator and sr_calculator extract trend and support/resistance information. sr_interaction tracks price behavior around levels; a touch is different from a break or rejection.

02 ICT and indicators

ict_engine analyzes structures such as order blocks, FVGs and liquidity. RSI, MACD, ATR and other indicators are consumed in timeframe context rather than treated as unconditional trade commands.

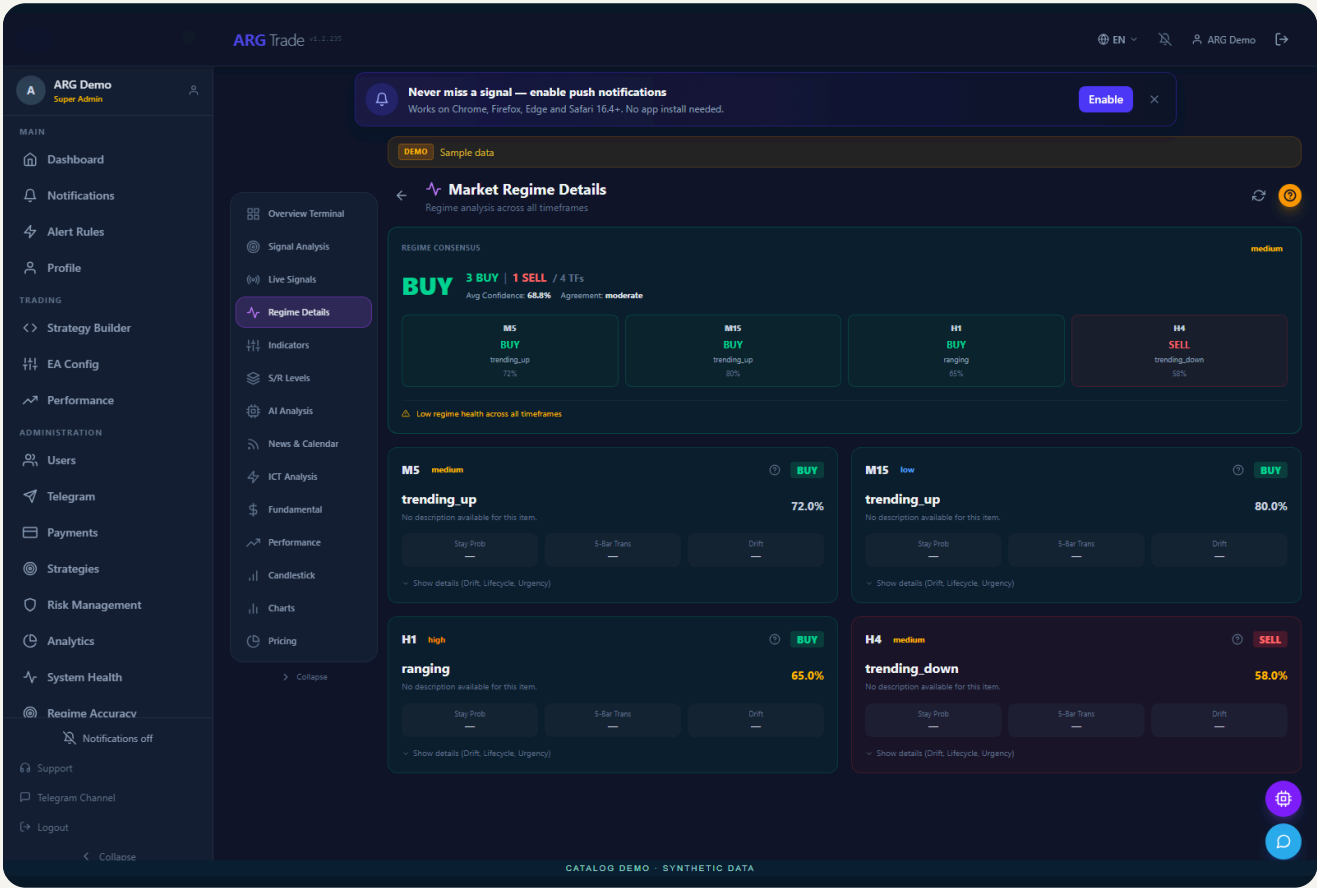
03 Context and explanation

Session, volatility, news, correlation and fundamental analysis enrich context. Separate scores and factors help operators understand a wait state or a rejected entry.

01 / MARKET REGIMES AND DIRECTIONAL CONTEXT

Market behavior before strategy selection

Regime detection converts market state into information for display, filtering and strategy evaluation.



Actual application interface, captured locally with synthetic data.

01 Multiple methods in the repository

The classifier includes GMM, HMM and hybrid methods, while ArgRegimeClassifier contains separate training and validation paths. The presence of an algorithm does not establish which model is deployed.

02 Output semantics

Regime identifiers, names, confidence, directional context and transition information can be shown per timeframe. Confidence is not a guaranteed trade win rate or profitability probability.

03 Evaluation and stability

Accuracy, drift and model-status surfaces expose quality and changing data behavior. Historical training figures are not presented as product performance without their data, window and independent evaluation.

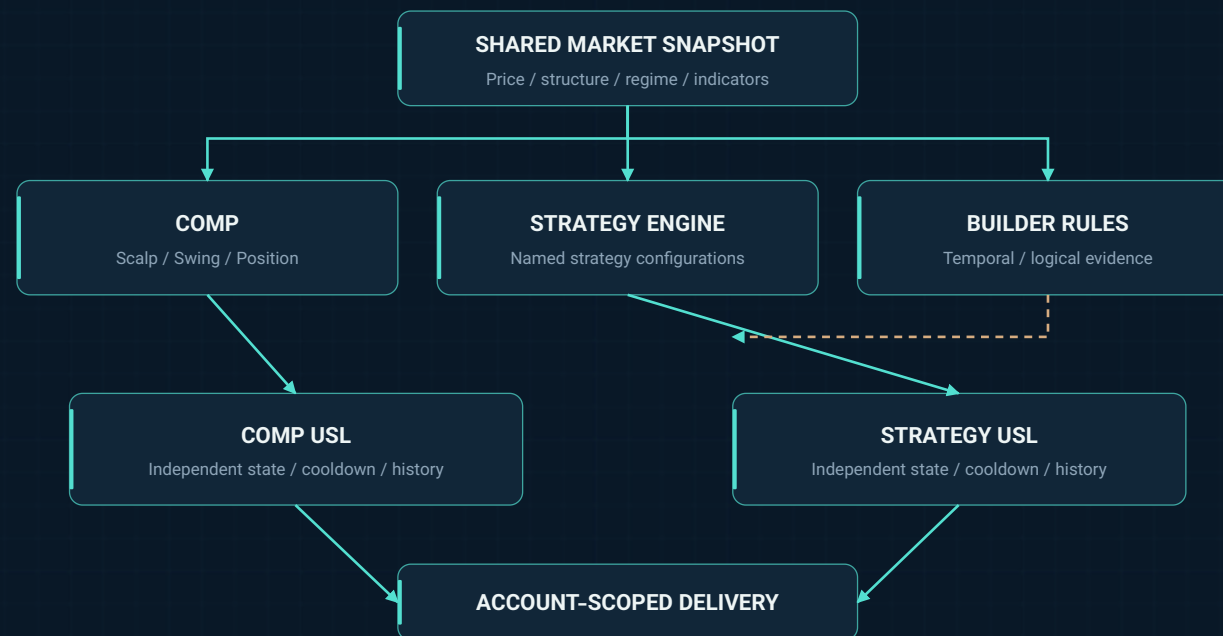
02 / COMPREHENSIVE ENGINE AND THREE HORIZONS

Three horizons. One market context.

Comp gathers analytical data and produces proposals, constraints and decision state for Scalp, Swing and Position horizons.

SIGNAL INTELLIGENCE / INDEPENDENT PRODUCERS

A shared context feeds different strategies without erasing their state or policy ownership.



USL disabled: production dispatch is blocked. Builder rules contribute evidence; they are not standalone EAs.

Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Separate horizons

Horizon policies come from CompML_Scalp, CompML_Swing and CompML_Position. The parent orchestration row does not replace each horizon's policy.

02 Candidate versus dispatch

A signal may be retained for diagnosis without dispatch permission. Comp requires explicit `dispatch_allowed=true` plus valid entry, stop and target ordering.

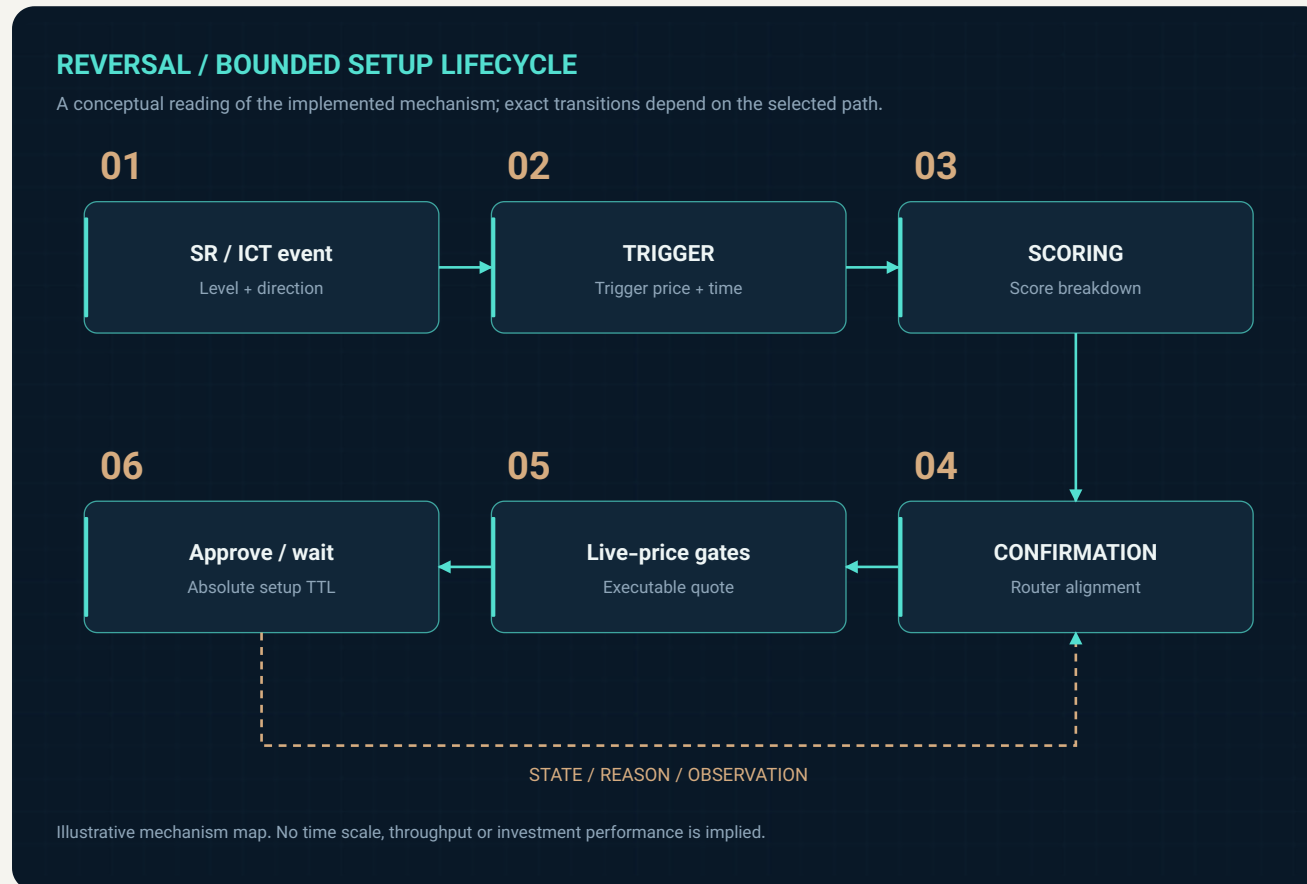
03 Decision trace

Direction, time, horizon, factors and block reasons connect decisions to delivery and outcomes. A BUY or SELL label alone does not establish that an order was executed.

02 / V2 REVERSAL LIFECYCLE

From reversal event to eligible signal

The reversal path is a bounded multi-stage lifecycle separating a level event, scoring and entry timing.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Trigger

Support/resistance rejection and supported ICT-zone events can originate a candidate. Level, direction, trigger time and trigger price travel with the signal.

02 Scoring and confirmation

V2Signal retains the decision phase, score breakdown, entry quality and abort reason. A candidate passes through contextual scoring and confirmation so acceptance, waiting and rejection remain explainable.

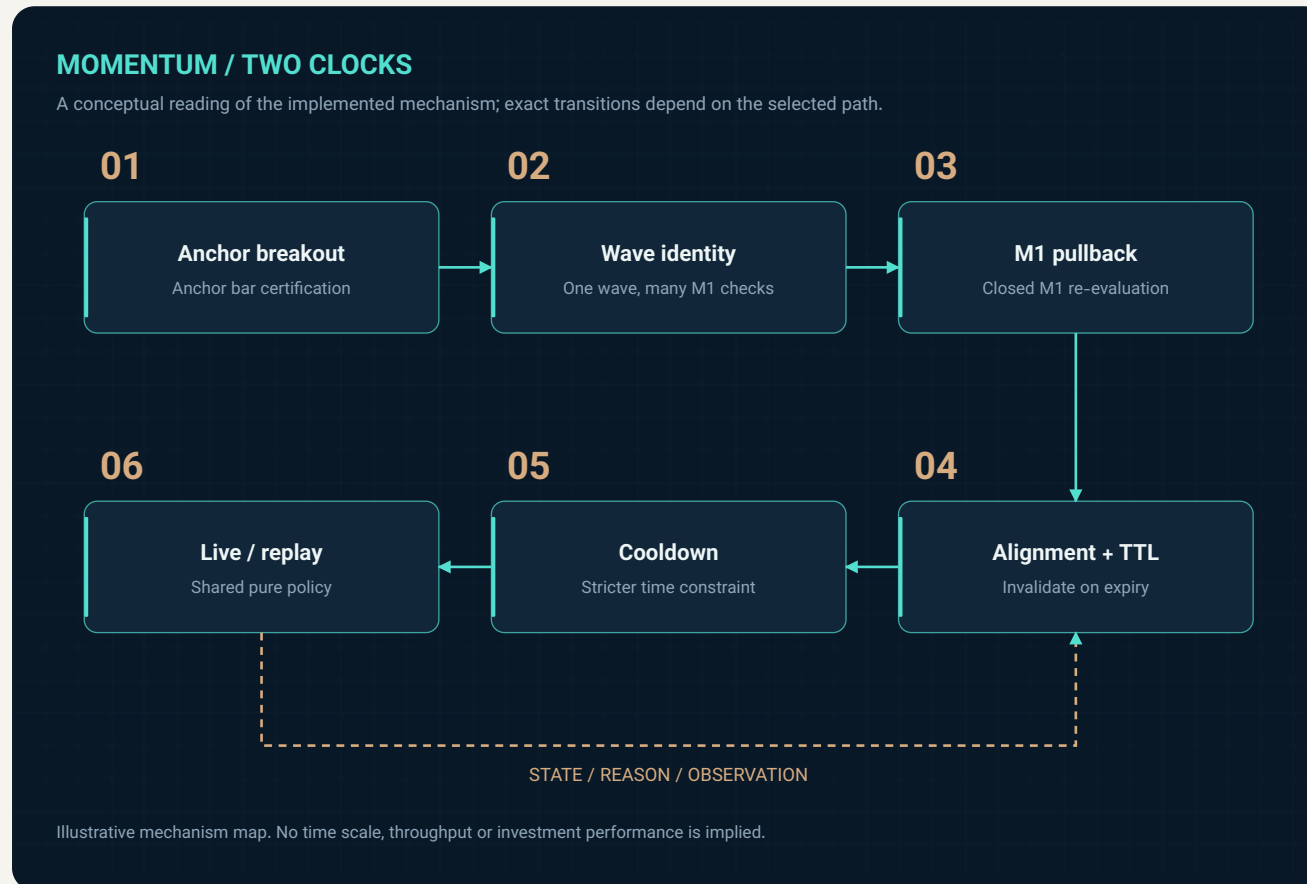
03 Lifetime and executable price

Router-alignment waits and unavailable executable quotes have bounded clocks. The absolute setup deadline must not restart whenever timeframe alignment changes.

02 / MOMENTUM AND WAVE POLICY

Identify the wave. Qualify the entry.

Momentum is distinct from reversal and explicitly manages breakout timing, entry location and wave consumption.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Two time bases

A breakout is certified on the anchor timeframe. In m1_pullback mode, entry location is re-evaluated on each closed M1 bar; repeated M1 evaluation must not create new waves.

02 TTL and cooldown

Alignment changes or TTL expiry invalidate wave permission. Effective cooldown uses the stricter duration from anchor-bar cooldown and the central M1-based dispatch interval.

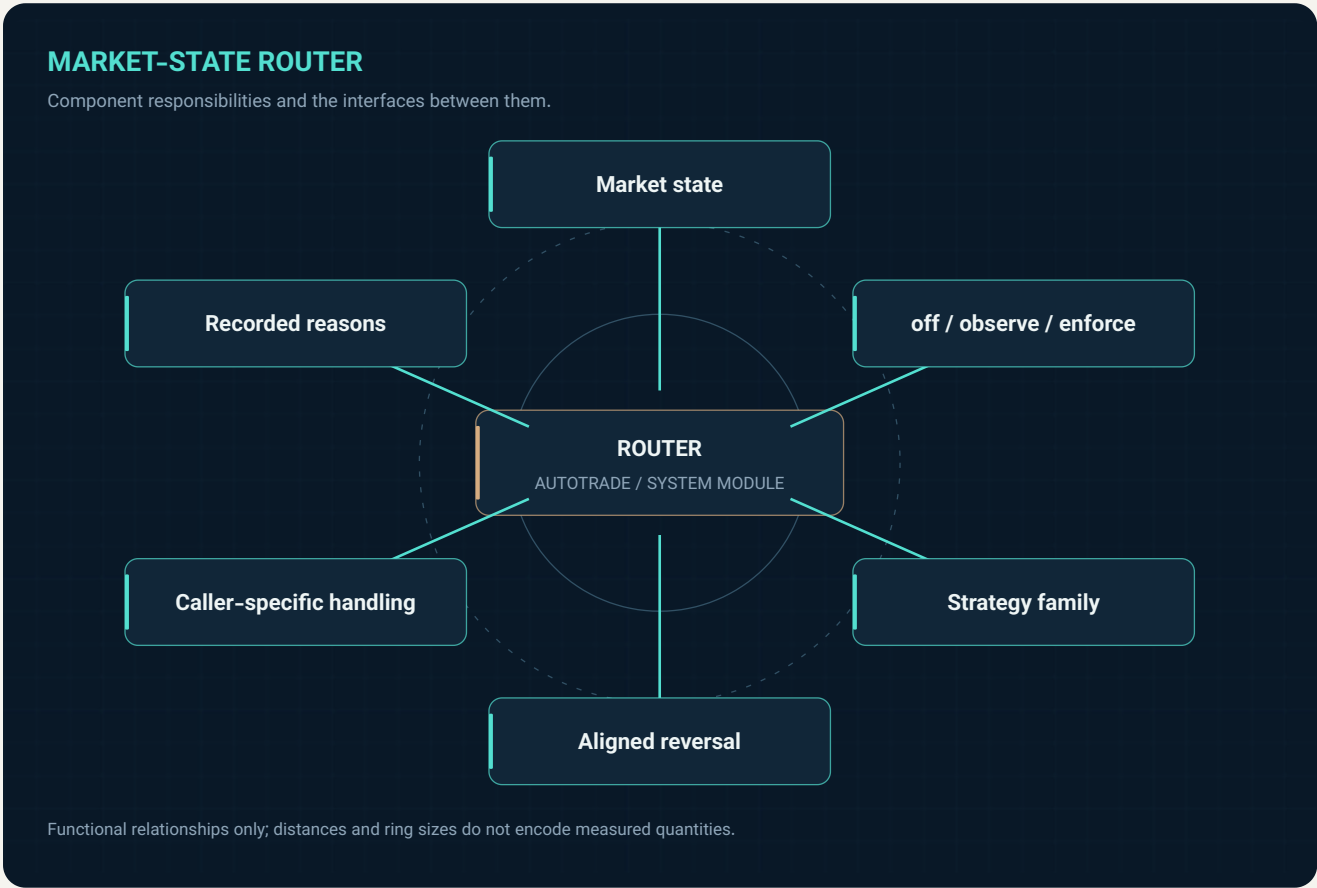
03 Replay consistency

momentum_execution_policy provides pure functions shared by live and replay paths. Order type and entry price must follow that contract so replay does not invent opportunities unavailable in live execution.

02 / MARKET-STATE ROUTING

Route strategies by market state

The router resolves which lane and direction may enter; observation differs from enforcement.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Market states

ONE_WAY, RANGE, TRANSITION and UNCERTAIN resolve into reversal and momentum permissions. A one-way trend can retain aligned reversal entries while restricting countertrend fades.

02 off / observe / enforce

off has no execution effect; observe records and compares suggested permissions; enforce lets the caller reject entries outside the enabled scope.

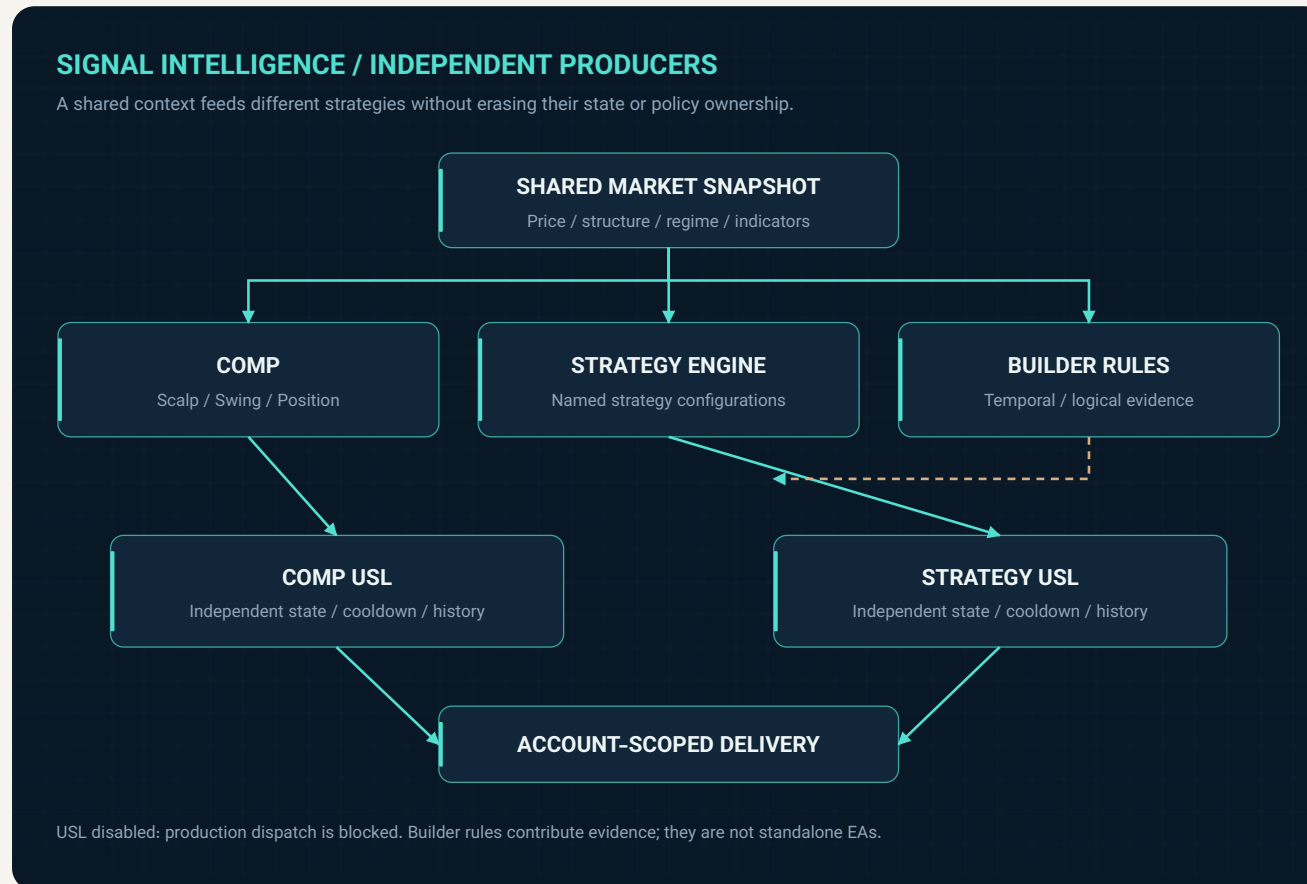
03 Insufficient evidence

Unknown state, low confidence and observer failure have explicit behavior. This router is not universally fail-closed and must not be described as equivalent to the hard feed-validity gate.

02 / UNIFIED SIGNAL LAYER

The final control layer for each engine

USL shares signal-geometry and quality policy while isolating each engine's mutable runtime state.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Two isolated instances

Comp and StrategyEngine have separate cooldown, history and dispatcher instances. Agreement may add positive evidence; an opposing producer must not gain arbitrary cancellation authority.

02 Final geometry

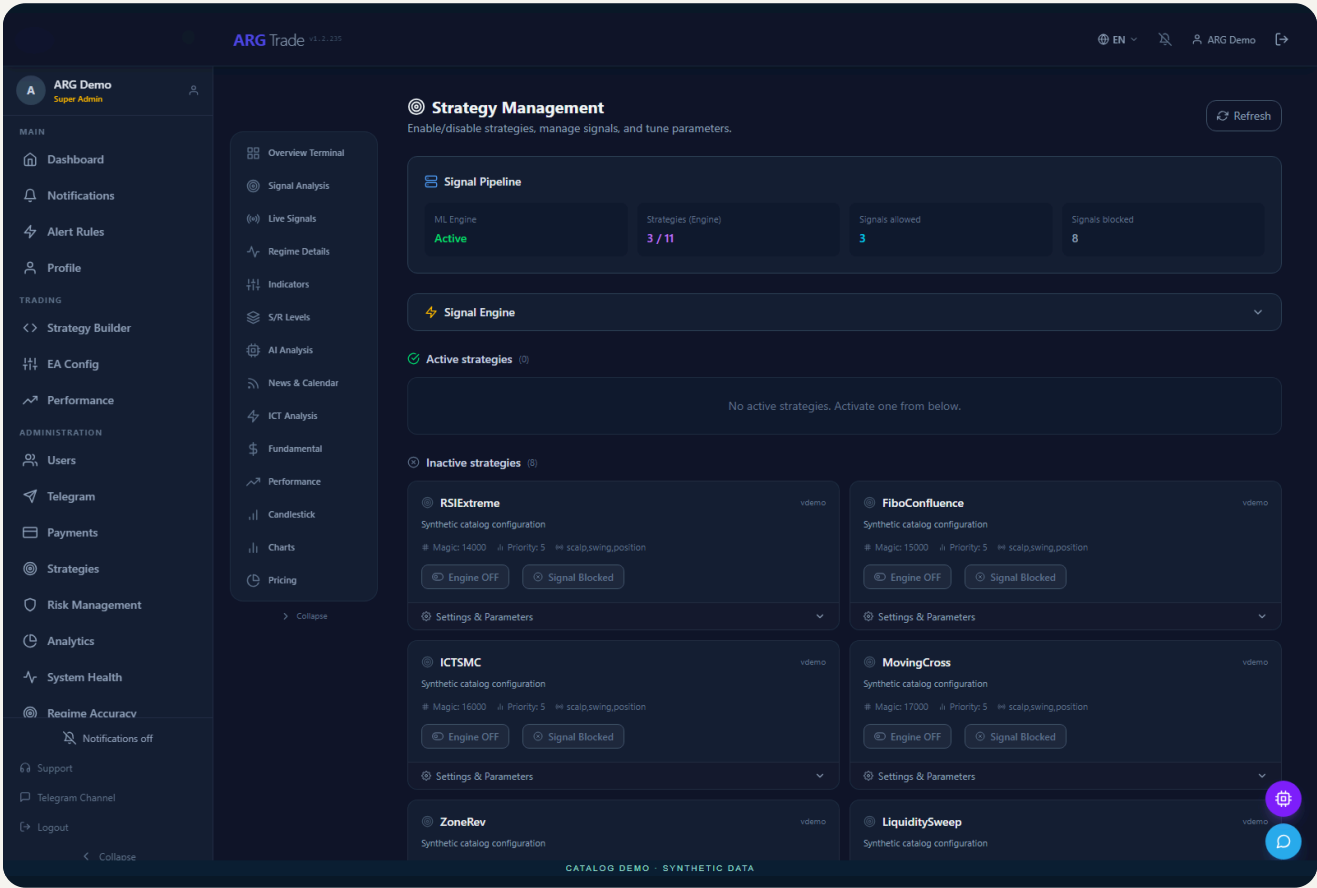
Final SL/TP follows the producing strategy and horizon policy. Direction and level ordering, cooldown, reversal gap and quality checks apply on the final path.

03 Safety switch

Disabling USE_UNIFIED_SIGNAL_LAYER does not create a direct-dispatch bypass; it blocks production signal paths. Actual thresholds and activation must be read from authoritative settings.

Distinct strategies. Explicit policies.

The Registry and Engine manage multiple trading methods with independent parameters, identity and state.



Actual application interface, captured locally with synthetic data.

01 Present strategy families

Source includes RSI Extreme, Fibonacci Confluence, ICT SMC, Moving Cross, Zone Reversal, Liquidity Sweep, Session Breakout and Trend S/R. A source file does not prove activation for a customer account.

02 Operational controls

The registry centralizes registration, enable/disable and parameter access. Strategy settings distinguish engine participation, signal delivery, horizons, priority and associated policies.

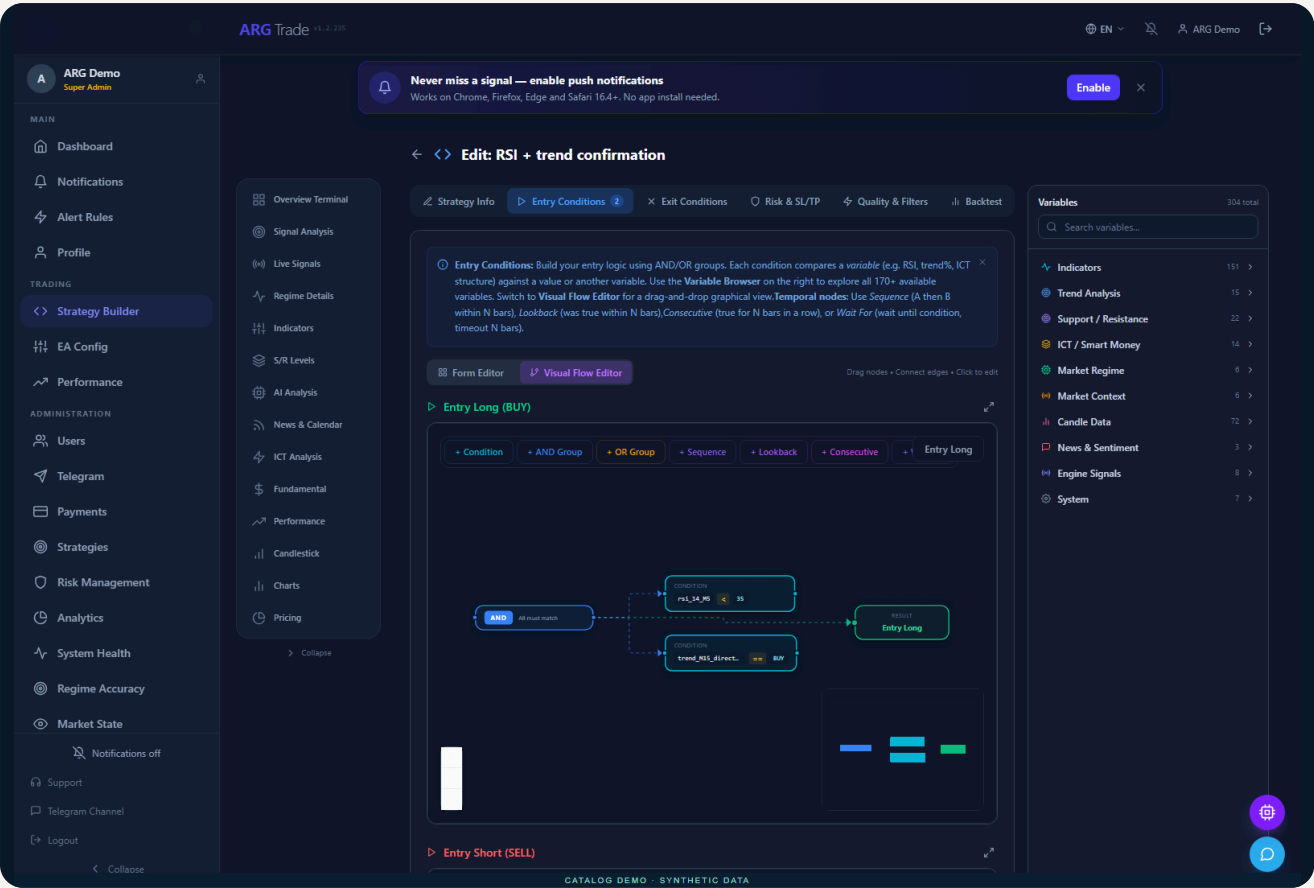
03 Per-method evaluation

Each method must be evaluated with its own entry/exit definition, costs, data and risk limits. This catalog makes no shared win-rate claim or claim that all strategies are active.

02 / VISUAL STRATEGY BUILDER

Make strategy logic visible

Trading rules are authored as condition trees linked to real analysis-pipeline variables.



Actual application interface, captured locally with synthetic data.

01 Logic and variables

AND/OR groups and leaf conditions reference indicator, trend, level, ICT, regime, context, candle, news and system variables. The registry supplies names, types, descriptions and parameters.

02 Temporal conditions

sequence, lookback, wait_for and consecutive express event relationships over time. Historical access must be supported by the resolver; not every snapshot variable has retrievable history.

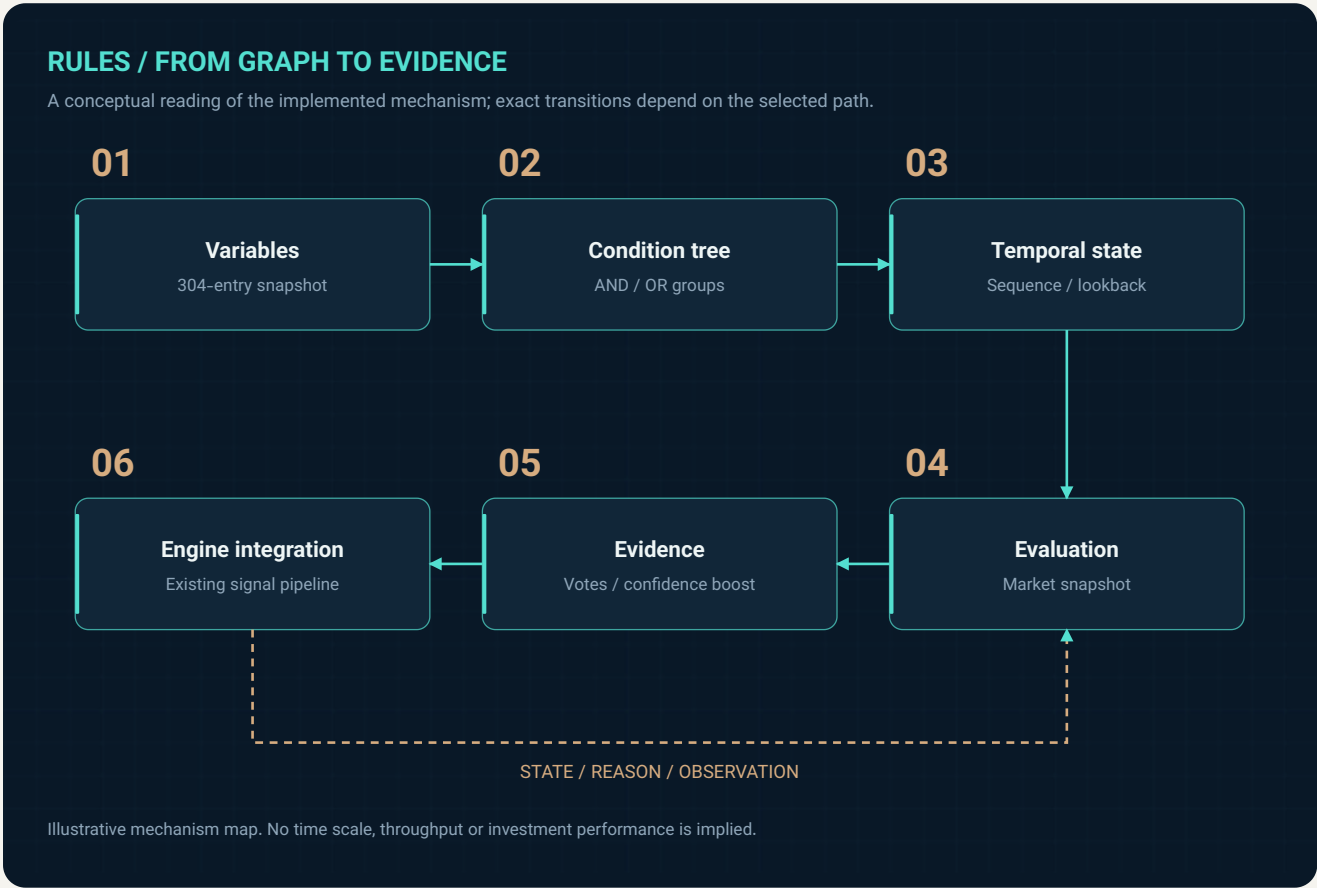
03 Authoring lifecycle

The interface and API support creation, editing, cloning, evaluation, activation and backtesting. The catalog example is a draft strategy using synthetic data.

02 / EXECUTION OF AUTHORED RULES

How does a visual rule run?

A visual rule becomes meaningful through evaluation and central policy; drawing a graph is not a market order.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Loading and priority

The integration layer loads enabled strategies from the database in priority order and checks the allowed horizon. The evaluator resolves variable values and the condition tree.

02 Integration result

The inspected integration returns matches, direction votes and confidence boosts, with per-strategy and total caps. This path is not a separately installed independent EA.

03 Temporal validation

The state store tracks temporal-rule state. Draft tests and backtests must preserve chronological order and data availability at the decision time.

02 / STRUCTURAL STOP AND TARGET CONSTRUCTION

Stops and targets with clear provenance

Stop distance derives from signal structure and allowed policy, then must agree with executable prices and broker constraints.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Level selection

smart_levels consumes signal structure, respected levels and an ATR floor. Final prices and reward/risk use the actual distance from entry.

02 Clamp or refuse

The current distance cap can clamp a stop or reject a candidate in refuse mode. Clamping or ML selection may move the stop inside the original structural level; provenance must preserve the adjustment.

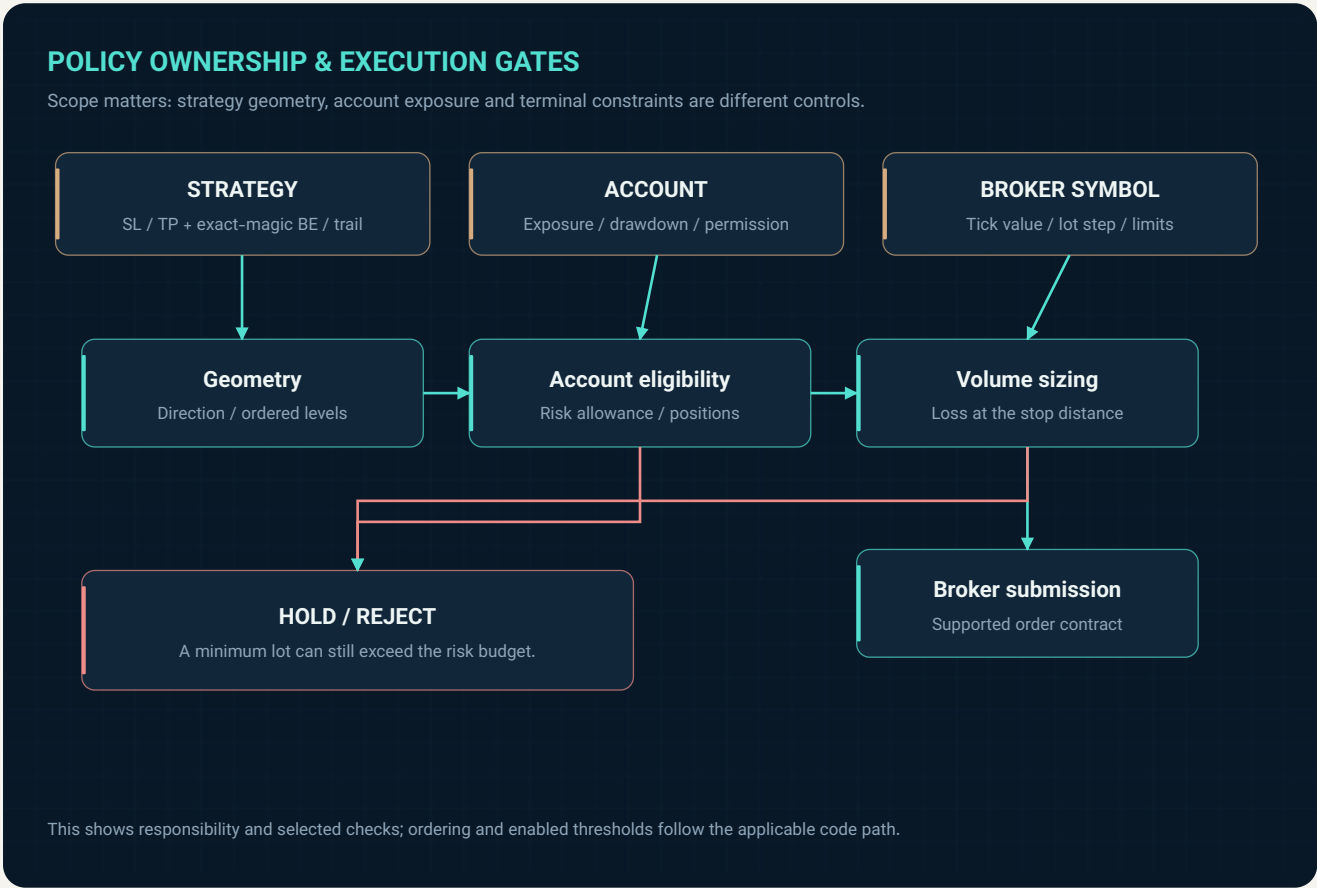
03 Analysis versus execution

Smart SL Quality evaluates outcomes; the stop resolver constructs geometry; the EA performs final execution checks and order handling. These are distinct responsibilities.

02 / RISK CONTROLS ACROSS LAYERS

Risk has an owner at every boundary

Account risk, strategy geometry and broker execution constraints are separate control layers.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Policy ownership

strategy_configs owns geometry and exact-magic BE/trailing policy. risk_profiles governs account exposure and drawdown. Global, profile and account settings must be interpreted at the correct scope.

02 Pre-execution checks

Relevant paths check trade permission, position count, direction, risk allowance, spread and symbol constraints. Administrative surfaces expose emergency stop and account state.

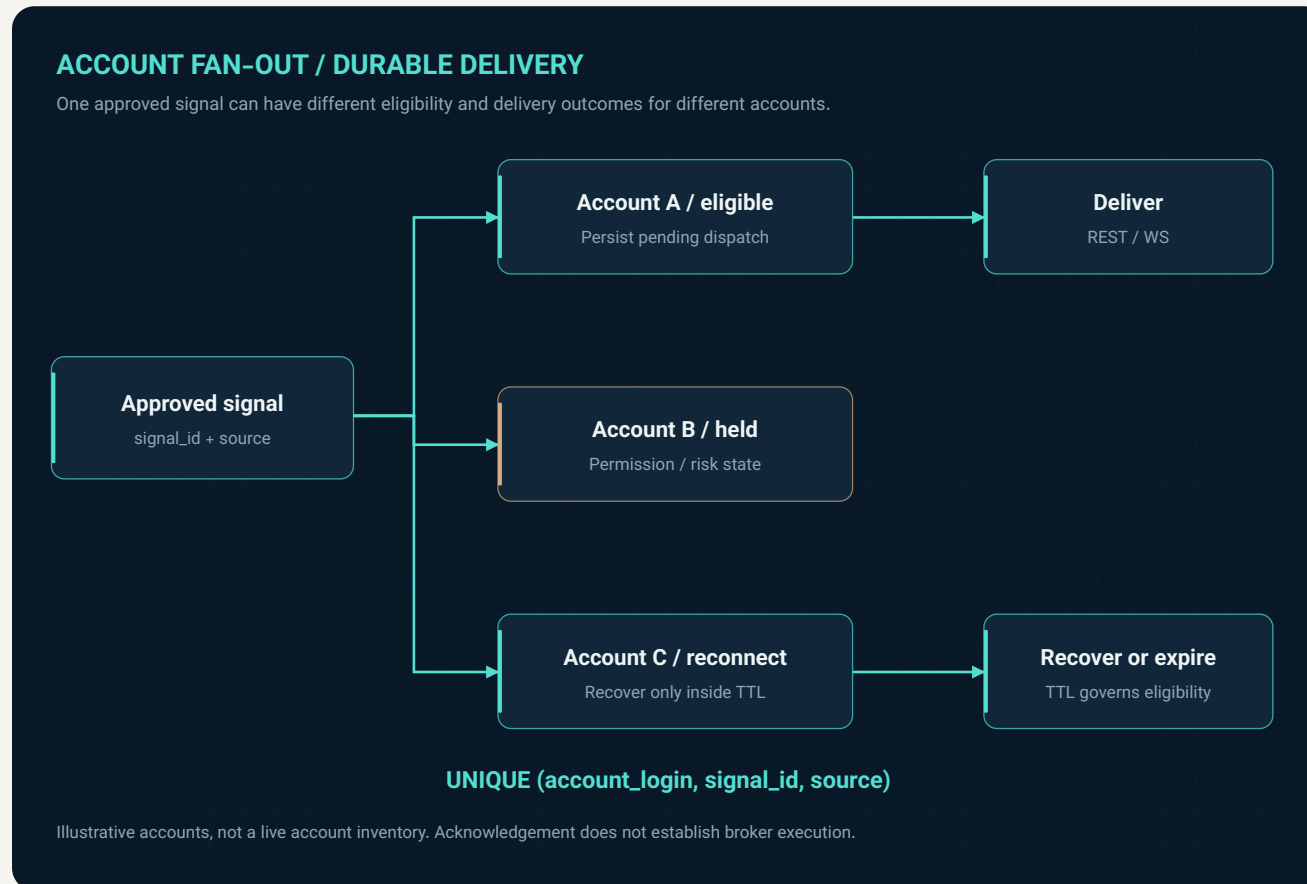
03 Actual volume

The EA checks volume against broker specifications and estimated loss at the stop distance. Minimum volume and volume step can make a trade inadmissible; the smallest lot is not necessarily within budget.

03 / DURABLE PER-ACCOUNT DELIVERY

Every account has its own delivery path

An approved signal becomes a separate dispatch record for each eligible account.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Durable queue

signal_dispatch_log retains pending delivery instead of relying solely on short-lived cache. The unique account_login, signal_id and source tuple prevents duplicate queue insertion.

02 TTL and readiness

Reconnection can recover only signals still within TTL. Transport heartbeat is distinct from EA execution readiness. Some incomplete legacy contracts retain compatibility through a counted legacy_fail_open path.

03 Acknowledgement and outcome

Dispatch, receipt, execution rejection and trade outcome are separate stages. A transport ACK alone is not proof of a broker fill; outcome attribution requires signal and account identity.

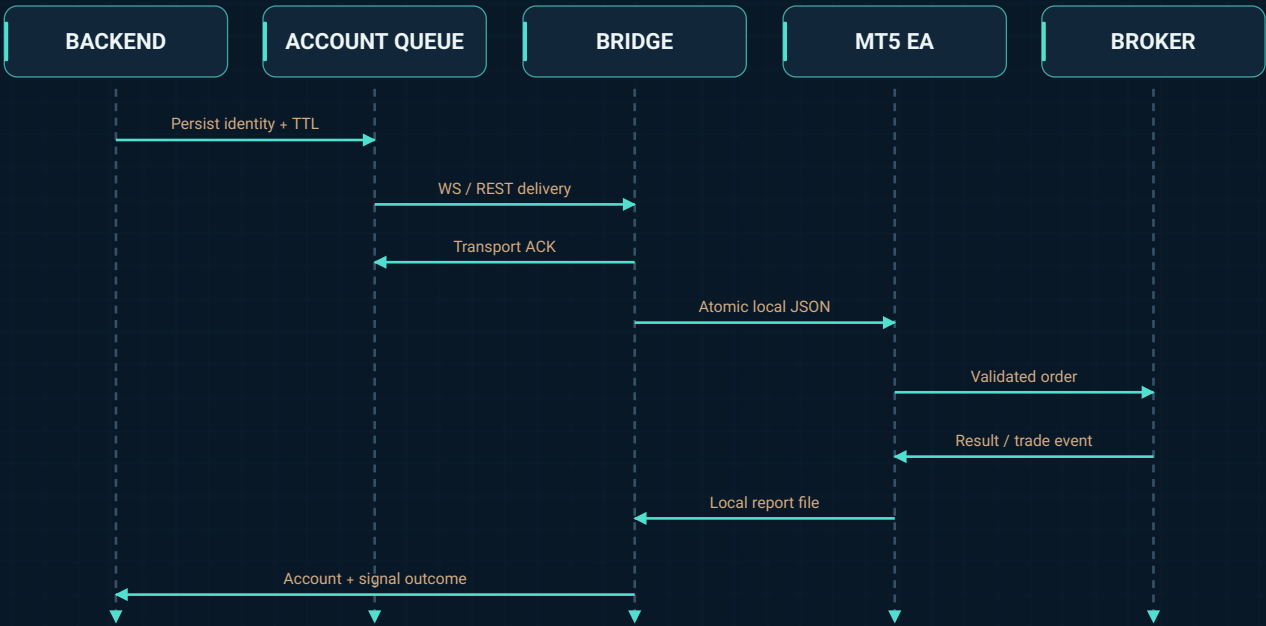
03 / PROTOCOLS AND COMMUNICATION CONTRACTS

From approved signal to broker response

Each connection has a defined role; real-time messages, durable records and local files are not interchangeable.

FROM APPROVED SIGNAL TO BROKER RESULT

Transport acknowledgement and broker fill are separate milestones.



EA validates freshness, geometry and volume before submission. Reconnect does not renew an expired setup.

Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 REST and WebSocket

REST/JSON carries data, configuration and reports. WS Dispatch uses separate inbound/outbound queues and push-on-change, reusing the signal endpoint’s account-scoped gating.

02 Redis and PostgreSQL

A Redis channel announces produced signals; PostgreSQL anchors durable records. The notification service uses Redis Streams and consumer groups for event processing.

03 Files and authentication

The Bridge uses atomic JSON replacement so the EA cannot read a torn file. The WS contract includes token-derived signatures and installation identity; this does not establish identical security properties for every API.

03 / PARSGOLD DESKTOP CLIENT

The connection between server and terminal

The client on the terminal computer manages the boundary between the network and MetaTrader.



Actual desktop interface; synthetic state or no loaded model.

01 Bridge

parsgold_client handles WS communication, version coordination, signals/configuration and local-file exchange. Stale or invalid messages are sanitized or replaced by a safe wait contract.

02 Installation and terminal lifecycle

pg_manager, pg_terminals and pg_common manage components, terminal discovery and shared utilities. pg_ui and the tray provide local status and controls.

03 Reporting and recovery

Trade reports, journals, health and installation identity support traceability. Mutex and process-identity checks protect defined paths from duplicate instances and incorrect activation.

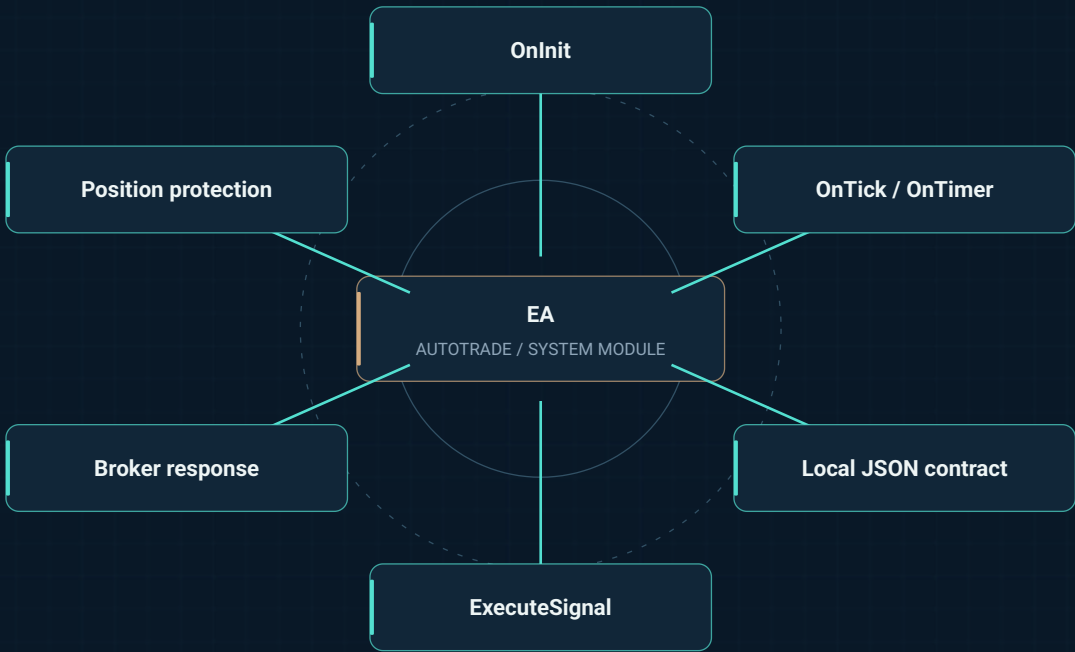
03 / THE EXPERT ADVISOR INSIDE METATRADER

What happens inside the MetaTrader EA?

ParsGold_AutoTrade is the execution EA; networking is delegated to the client so trade-management loops do not depend on HTTP response time.

INSIDE THE METATRADER EA

Component responsibilities and the interfaces between them.



Functional relationships only; distances and ring sizes do not encode measured quantities.

Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Startup and execution loops

OnInit prepares resources and managers. OnTick and OnTimer process price, risk and trailing. OnTimer checks file-based contracts and signals at the defined cadence.

02 Order handling

ExecuteSignal handles BUY/SELL and supported pending-order types. Identity, freshness, account/direction constraints, price geometry and volume are checked before broker submission.

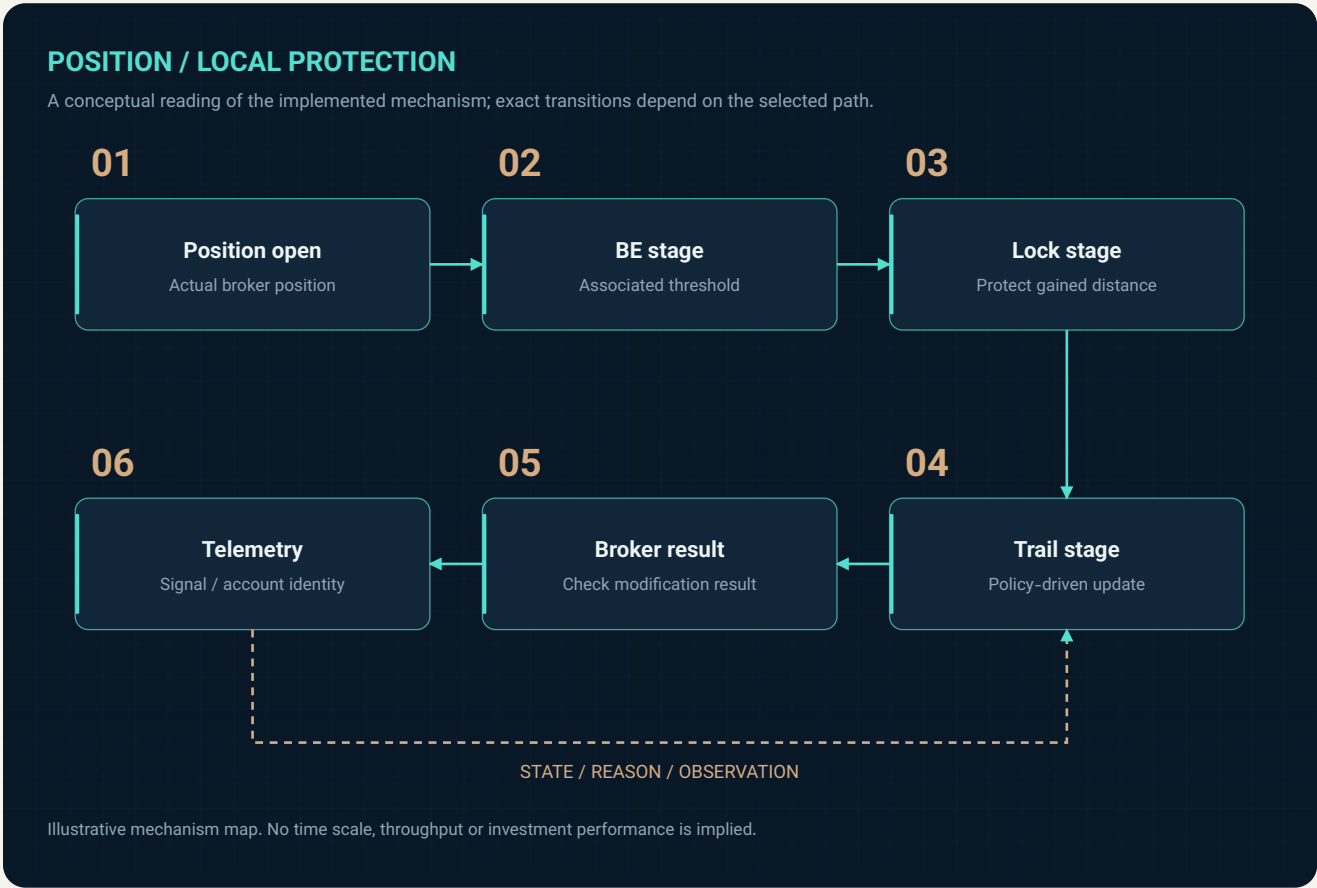
03 Actual integration

In this architecture the EA uses pg JSON files and the Bridge; a legacy description of a direct EA WebSocket does not describe this path. A source version is not proof of the installed binary version.

03 / POSITION HANDLING AND TRAILING

Position protection at the execution point

Post-entry management continues inside the EA using the policy associated with the strategy.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Protection stages

BE, Lock and Trail provide staged position protection. Thresholds, distances and activation follow the associated policy; one universal value does not suit every strategy.

02 Local execution

Trailing is separated from network latency. Pending orders have cancellation and state checks. Network loss and terminal shutdown are different operating conditions.

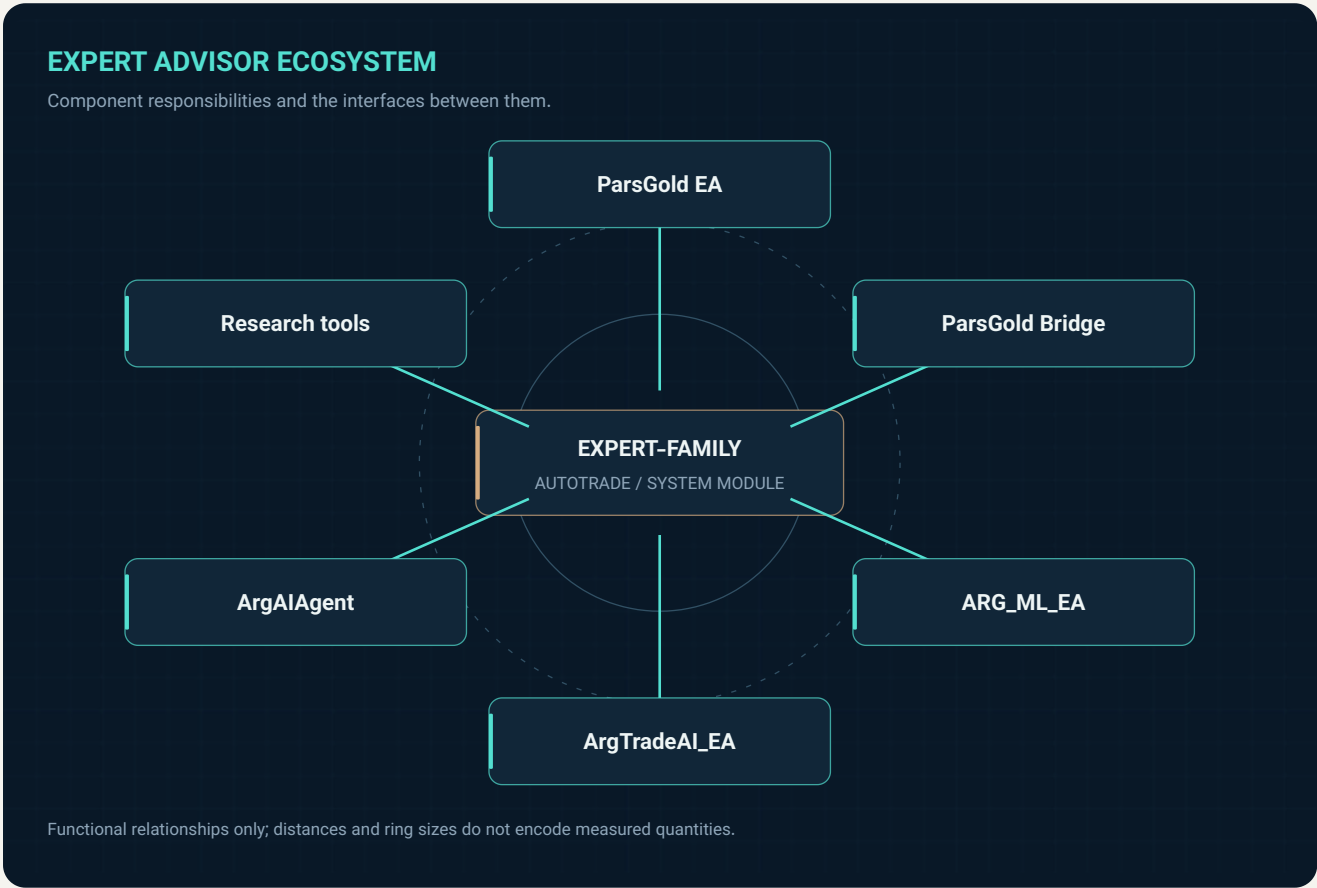
03 Outcome traceability

Trade events and position state are reported for signal attribution and exit-quality analysis. Stop changes, exits and partial closes must be interpreted through supported paths and actual broker responses.

03 / EXPERT ADVISOR FAMILY AND COMPANION TOOLS

The EA family and companion tools

The customer execution EA is distinct from data-collection EAs and research projects.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 ParsGold / ARG TradeClient

ParsGold source documents continuity of internal identifiers and contracts with the ARG_TradeClient family. The customer path focuses on backend-signal execution and order management.

02 ARG_ML_EA

This branch includes DataCollector, RegimeDetector, ShadowTrading and TradeTagging. Collection and shadow experimentation must be distinguished from actual customer orders.

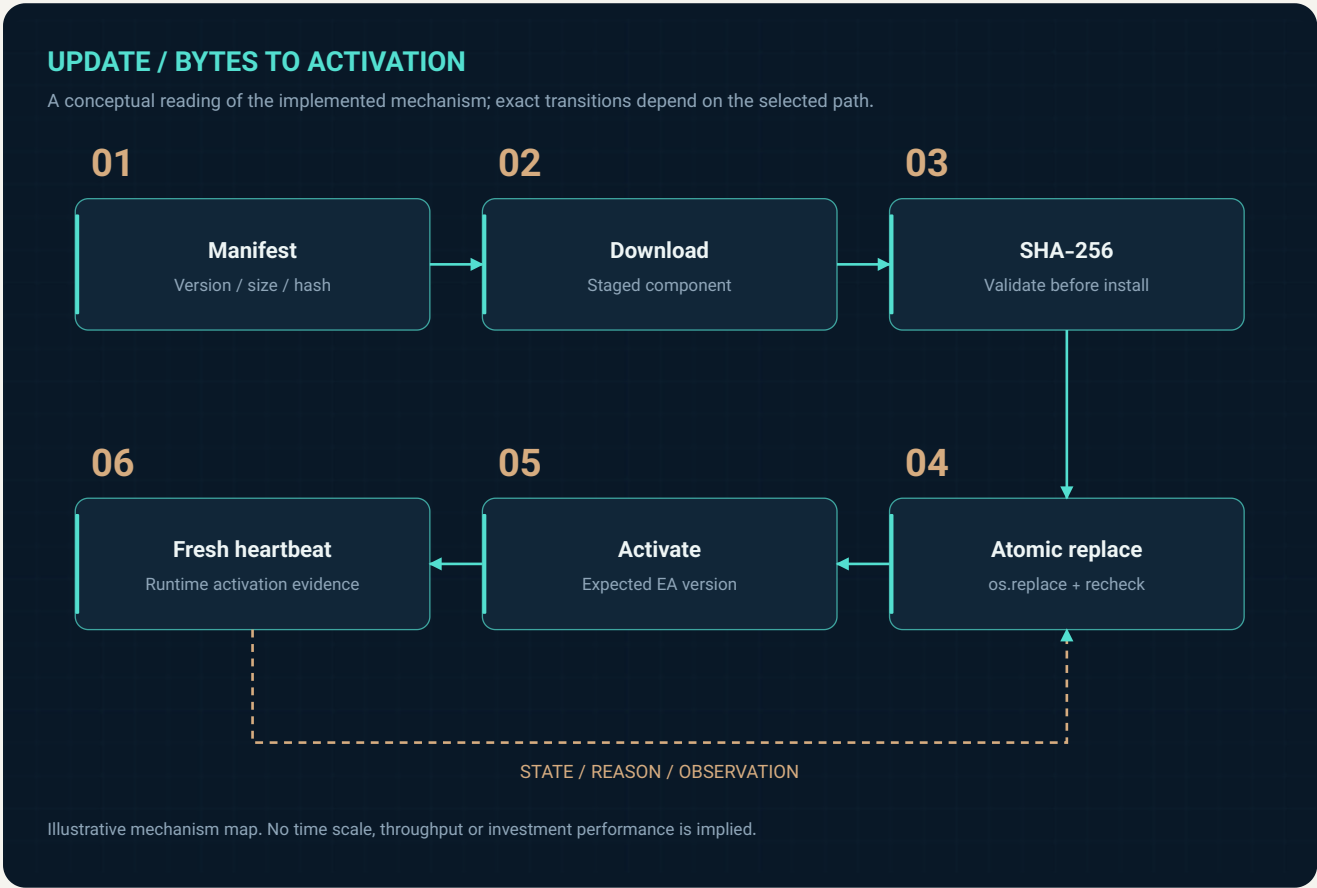
03 Historical sources

Recovered notes about legacy ARG_AutoTrade and adjacent projects provide history but do not justify assigning every historical capability to the current EA. This catalog identifies the source family for each claim.

03 / UPDATES AND ACTIVATION EVIDENCE

Installation and activation are distinct

Downloading an update is not completion; installed bytes and the running component must match the target.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Manifest and hash

pg_updater validates component manifests, versions, sizes and SHA-256. Downloaded bytes are checked before installation and after replacement.

02 Atomic installation

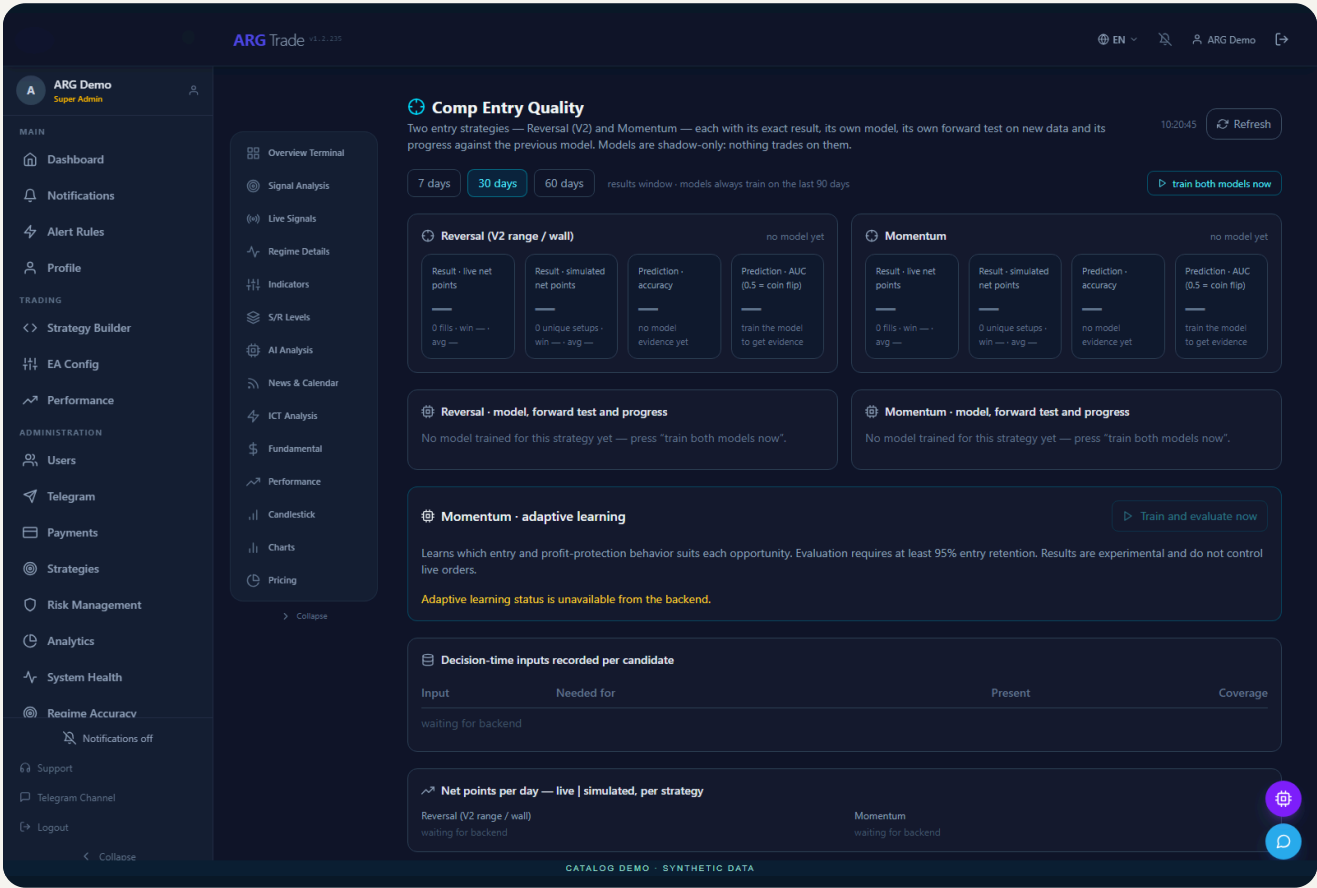
Validated staged bytes are installed through os.replace. Component state and attempts are recorded; an individual failure must not disappear behind a generic success message.

03 EA activation

EA activation evidence combines the expected hash with a fresh heartbeat from the expected version. Waiting, circuit and recovery states remain distinguishable in updater reports.

Observe and evaluate entry quality

In the inspected reference, Comp Entry ML predicts and records evidence without independently blocking trades.



Actual application interface, captured locally with synthetic data.

01 Per-lane models

Runtime loads lane-specific models from a registry and checks version changes. Feature vectors must match the model’s expected ordering and feature version.

02 Label provenance

Model label_source must match canonical execution attribution. Without a valid model the result is unscored, rather than an invented confidence value.

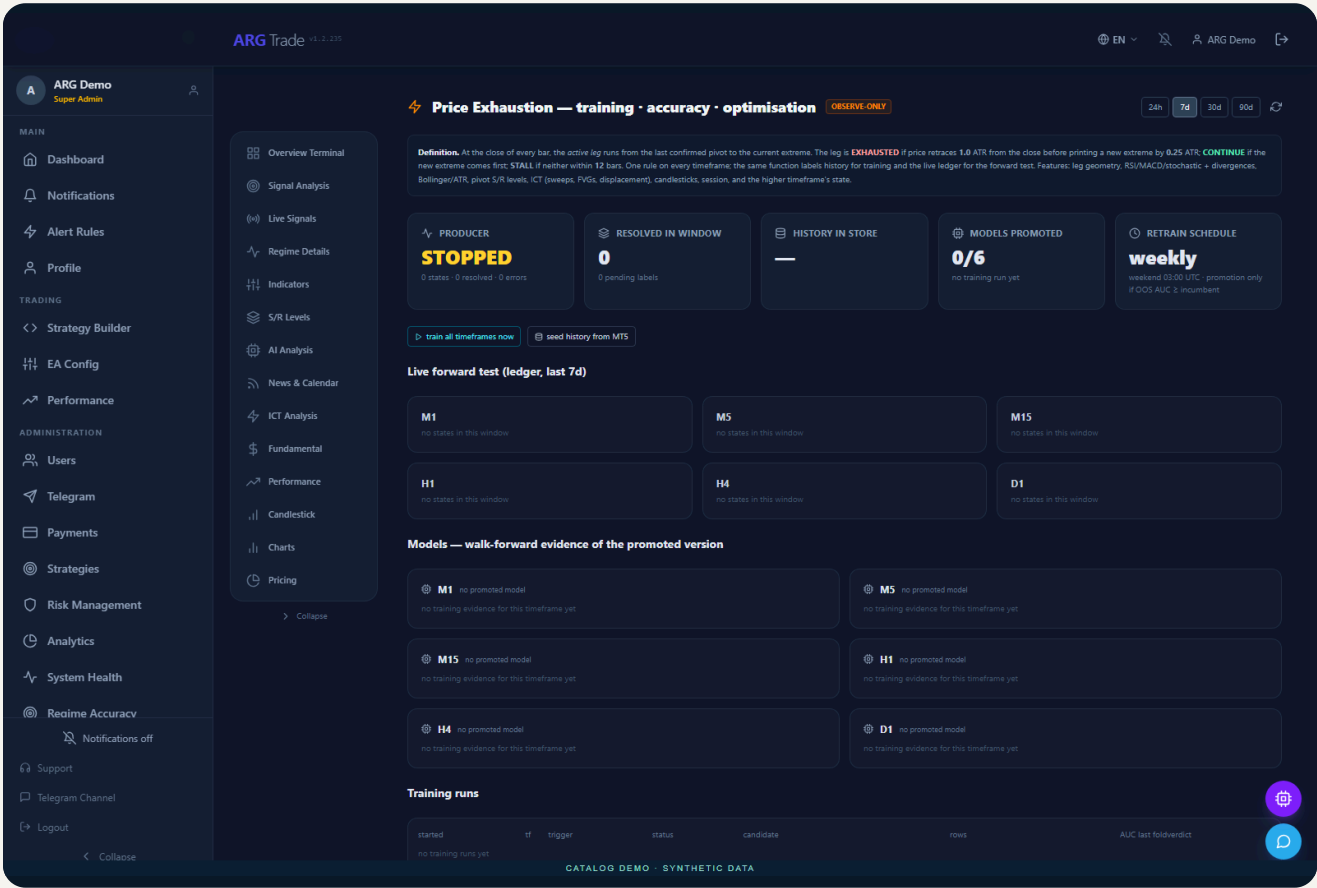
03 Authority boundary

The module declares PHASE=observe. Candidate scoring records features and probabilities; this catalog does not describe it as an enforced trade filter.

04 / QUALITY, EXHAUSTION AND POLICY EXPERIMENTS

Inspect the decision alongside the outcome

Diagnostic surfaces expose decisions, outcomes and differences between proposed policy and actual behavior.



Actual application interface, captured locally with synthetic data.

01 Exhaustion

Exhaustion modules and pages expose state, series, models, training and reports. An exhaustion score alone does not prove trading behavior changed unless its consumer is enabled.

02 Position Management

The analytical position_management module reports R progress, exit policies and conservative replay. This analysis does not itself send stop-change or close commands.

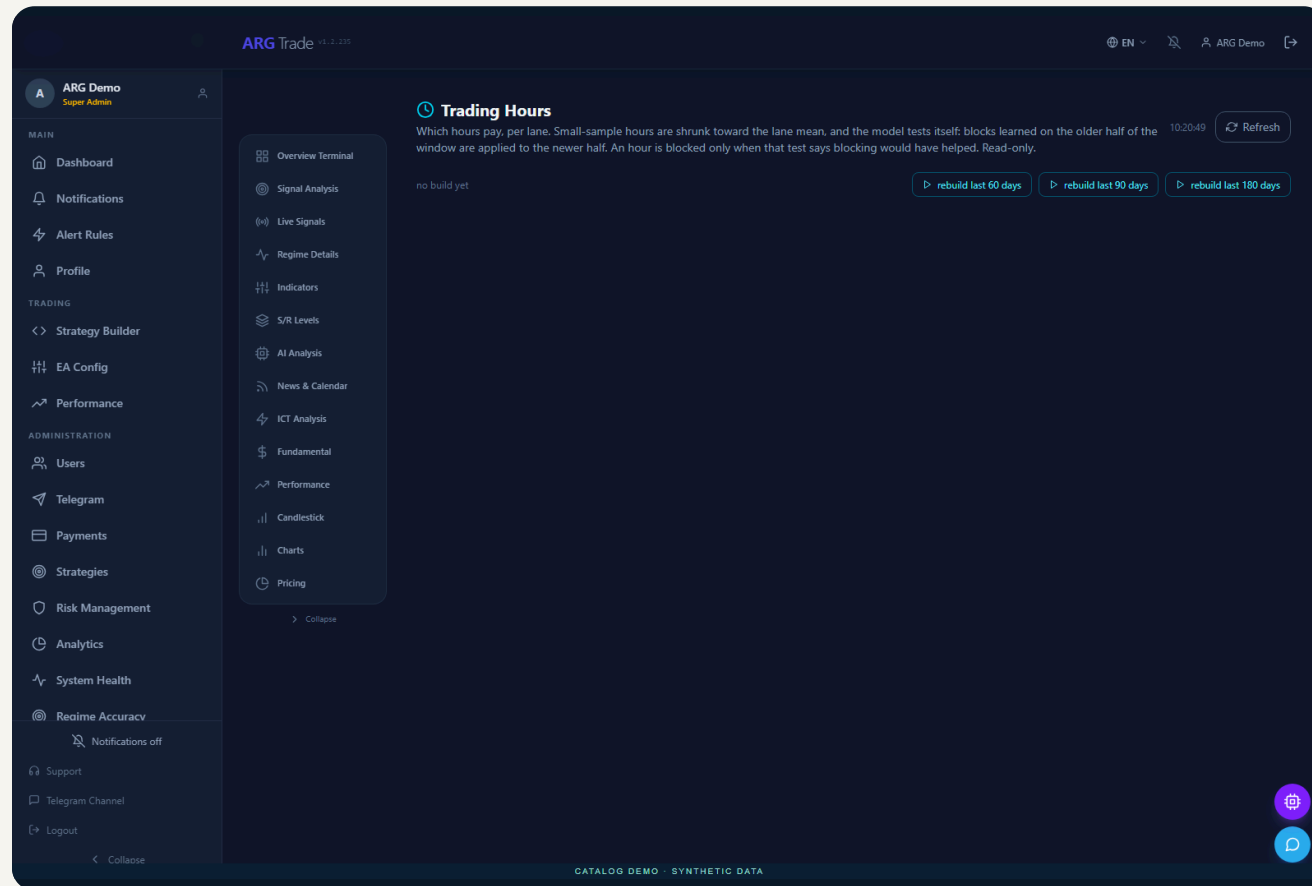
03 Stop and entry quality

Smart SL and Comp Entry surfaces expose evaluation and training evidence. Actual, shadow and replay results must retain distinct labels.

04 / TRADING HOURS AND TEMPORAL LEARNING

Evaluate timing on separated data

Trading-hours learning uses actual execution outcomes to construct testable lane-specific policies.



Actual application interface, captured locally with synthetic data.

01 Data and weighting

Closed fills are joined through explicit signal identity and fleet copies of one opportunity are aggregated. Shadow outcomes remain a separately labeled cross-check.

02 Out-of-sample check

UTC-hour statistics use shrinkage and minimum samples. Policy is fitted on the older half and evaluated on the newer half; a few poor trades in an hour do not justify blocking it.

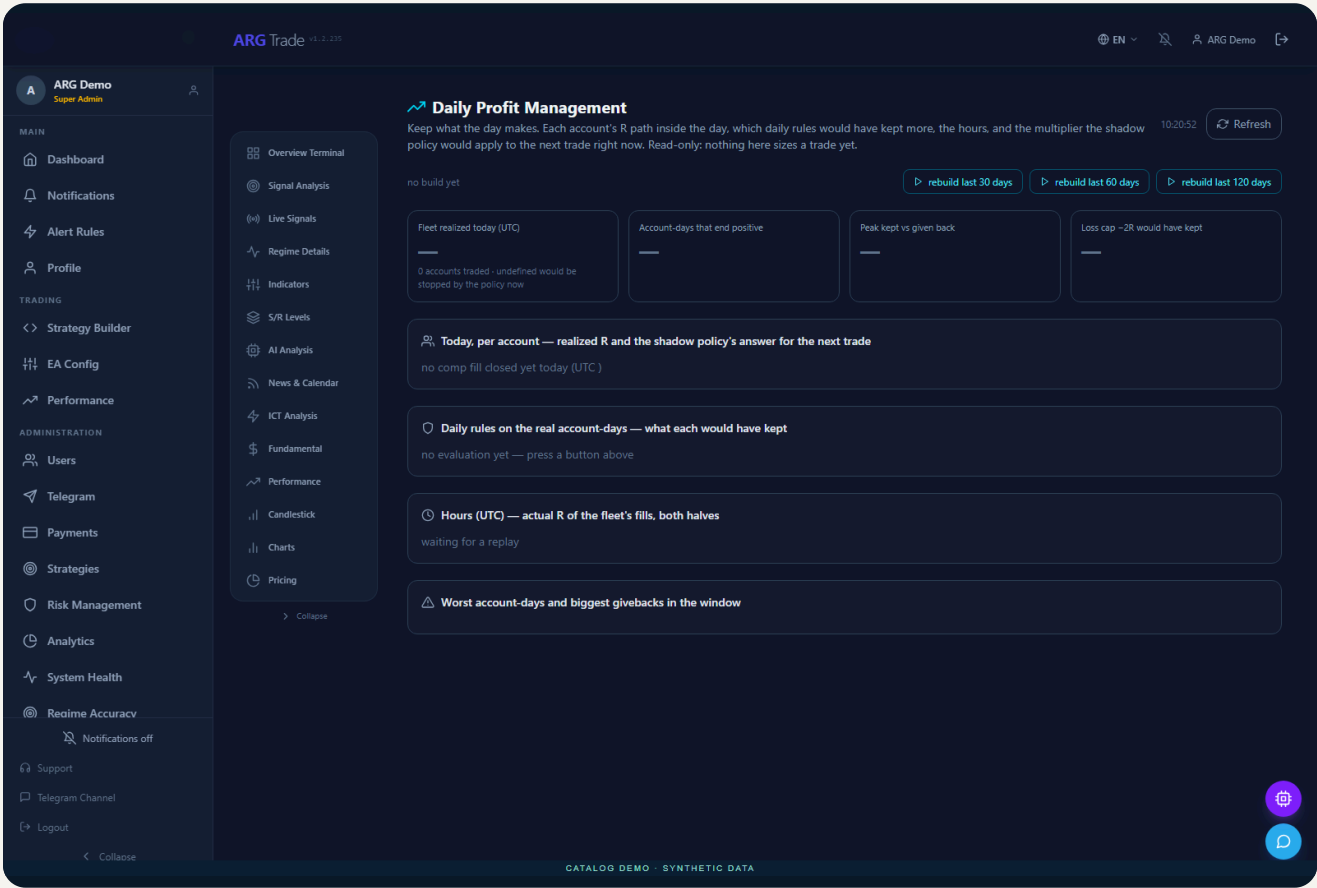
03 Enforcement

The learner and enforcement gate are separate. `th_enforce` and the relevant lane must be checked; a suggested 0.5 multiplier is not described as halving actual lot size without tracing the sizing path.

04 / DAILY POLICY AND PROFIT RETENTION

Daily policy with explicit authority

Daily Profit tracks each account’s daily state and proposed rule effects in R units.



Actual application interface, captured locally with synthetic data.

01 Actual daily path

Net closed-trade outcomes relative to opening risk, daily peak, close and giveback are tracked. Costs and execution provenance are part of the outcome definition.

02 Shadow and optional enforcement

Loss cap, soft cap, giveback and trade-cap policies can be analyzed. Phase-two enforcement is off by default in source and depends on dp_enforce and the account scope.

03 Effect and limitation

When properly enabled, a zero multiplier can stop account entries and an intermediate multiplier can scale risk_percent. These policy paths are defined as fail-open; they are not a loss-prevention guarantee.

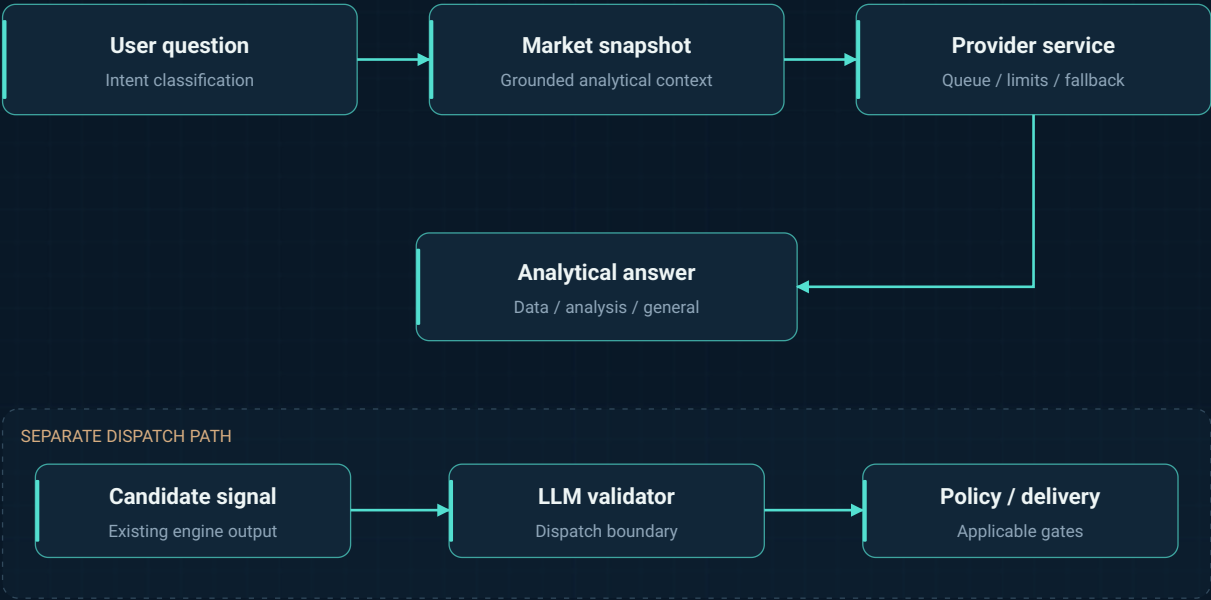
04 / AI SERVICES AND ANALYTICAL ASSISTANT

Market-grounded analytical assistance

Language models support reports, interpretation and defined validation paths; their role differs from order execution.

AI / GROUNDED INTERPRETATION & SEPARATE VALIDATION

The analytical assistant and dispatch-time LLM validation serve different purposes.



Assistant prose is not a raw broker command. Provider availability and enabled policy vary by deployment.

Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Provider infrastructure

The provider layer includes queues, rate limits, response statistics and fallback. Supporting several providers does not mean all are enabled or used simultaneously.

02 Web assistant

ai_assistant separates data, analytical and general intents and grounds responses in market snapshots. Model routing and quotas are explicitly defined.

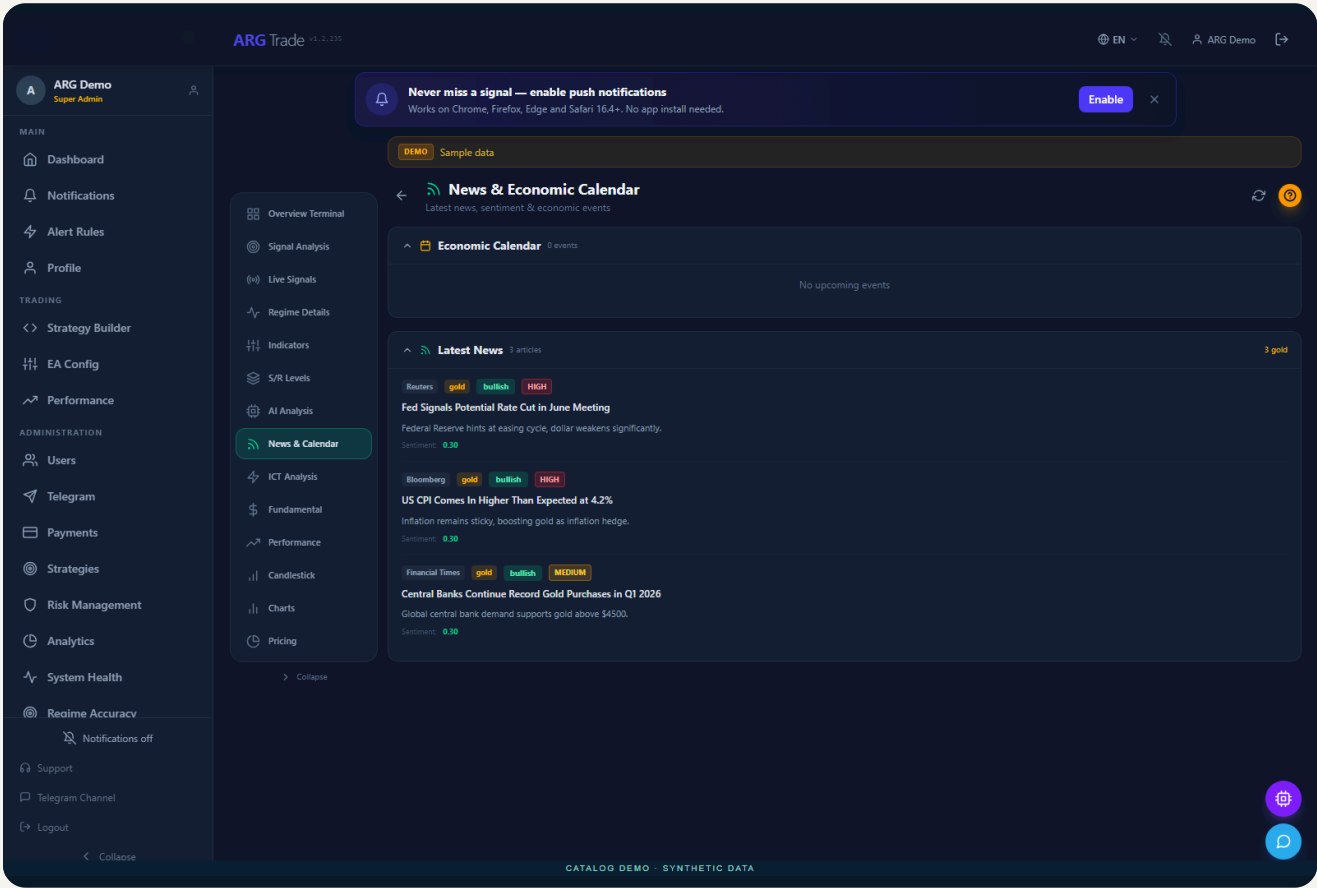
03 Signal validation

The LLM validator is invoked at defined dispatch sites, not for every raw USL output. An assistant’s textual response is not itself an approved execution command.

04 / NEWS, CALENDAR AND TELEGRAM SERVICES

News and events alongside technical context

Economic events and news add temporal and narrative context alongside technical analysis.



Actual application interface, captured locally with synthetic data.

01 News data

The backend exposes news, summary and calendar endpoints. ArgTradeAINewsService provides independent calendar ingestion and database caching. Source, time and status should remain visible with the content.

02 Telegram

Telegram service code and management APIs support messages, status and user interaction. A bot, a separate worker and the notification engine have different responsibilities.

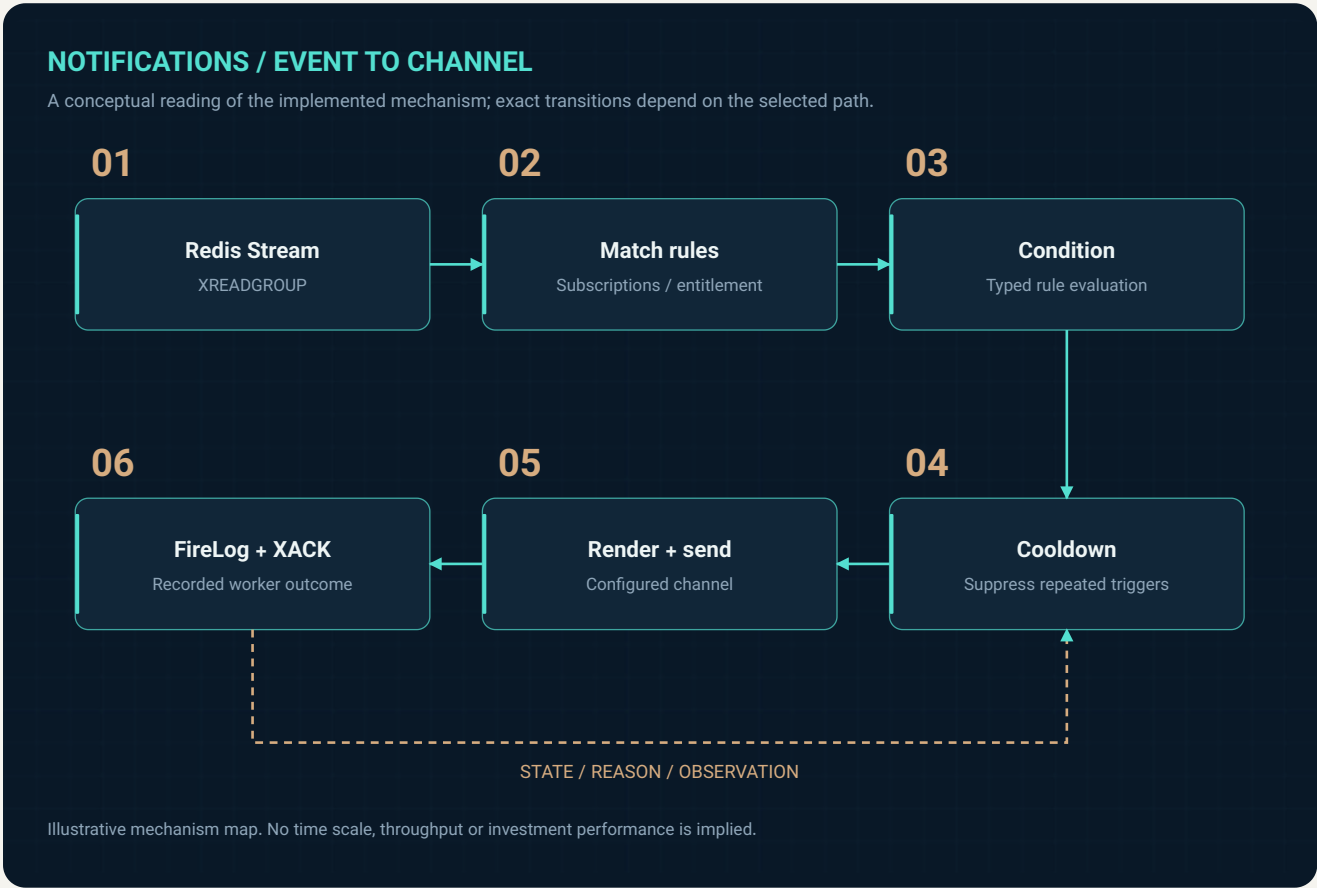
03 Product use

Calendar views, news categories, sentiment and analysis factors help explain market context. Event timing and content depend on providers; catalog screenshots are not current market data.

04 / RULE-BASED NOTIFICATION ENGINE

From a user rule to a traceable alert

Users define conditions, message templates and channels; events are processed by a separate service.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 **Processing lifecycle**

The worker reads events with XREADGROUP, locates rules, evaluates conditions and cooldown, renders messages, records FireLog and acknowledges with XACK.

02 **Channels**

Source contains Telegram, email, push, webhook and SMS channel implementations. Availability, quotas and providers depend on configuration and entitlement.

03 **Visibility and operations**

Rules, templates, subscriptions and delivery logs are separate resources. Matching a condition, successful transport and actual user receipt are different stages.

04 / BACKTESTING, REPLAY AND EVALUATION

Model, replay and live evidence

The repository contains several simulation and research paths; result validity depends on matching the tested path to the claimed behavior.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Testing components

data_manager, simulated_clock, signal_generator, trade_executor, pending_queues, engine and pro_engine serve different simulation responsibilities. BacktestIntelligence provides companion analysis tooling.

02 Outcome definition

Data window/provenance, spread, commission, entry type, intrabar stop/target ordering and exit policy belong with every result. Shadow, real-account and backtest outcomes are not interchangeable.

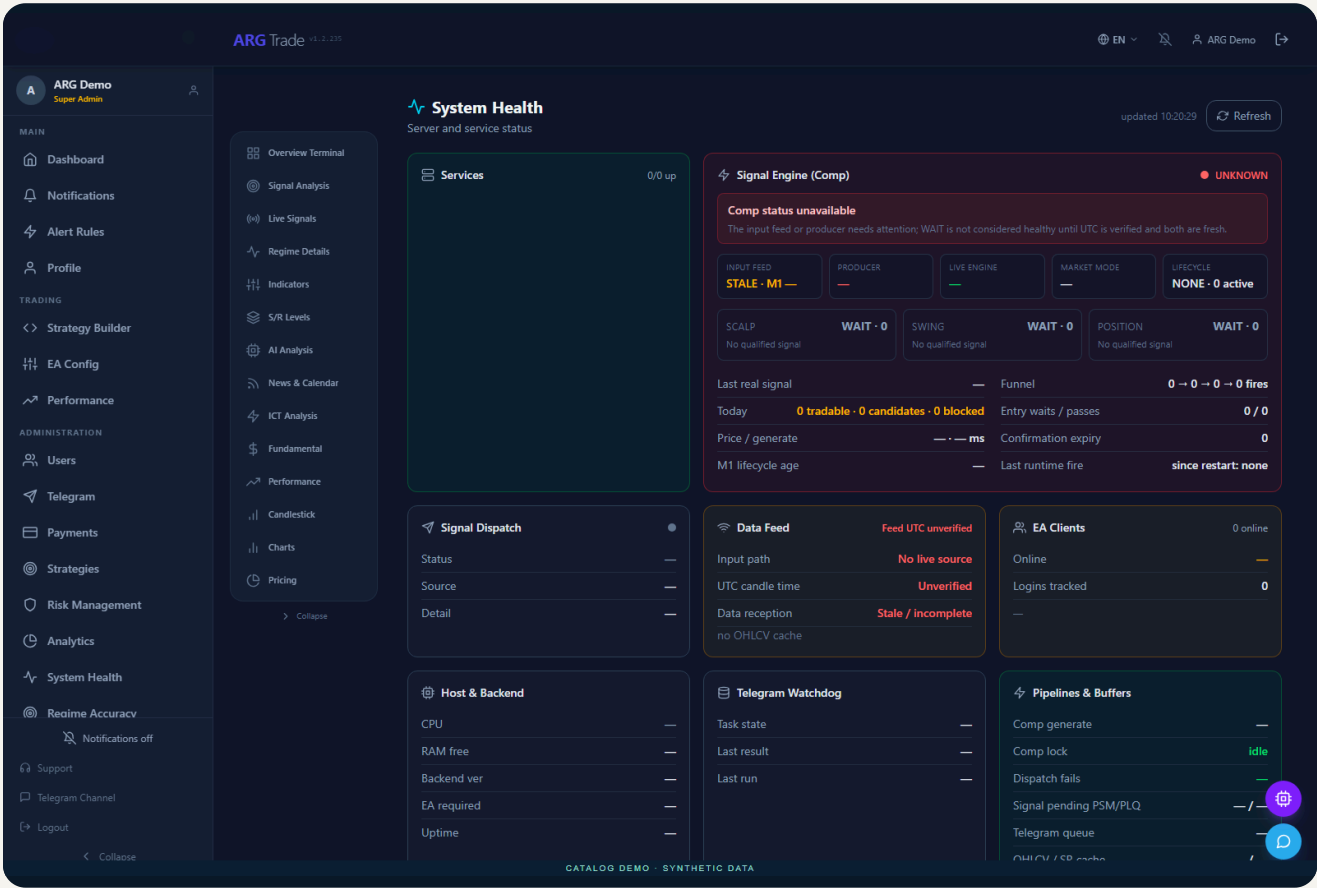
03 Claim boundary

This catalog project did not execute live trades or perform a new profitability study. Displayed charts and synthetic figures illustrate interface structure only.

04 / OPERATIONS AND OBSERVABILITY

Connection health. Readiness. Outcome.

An operational view must distinguish data, engine, transport and terminal execution health.



Actual application interface, captured locally with synthetic data.

01 Different health dimensions

/health, system-health, service status, EA Monitor and diagnostics cover different scopes. HTTP 200 or a WS connection alone does not establish trading readiness.

02 End-to-end traceability

Signal identity, account, source, version and timestamps link production, dispatch, ACK, execution and outcomes. Stale telemetry must be distinguished from fresh state.

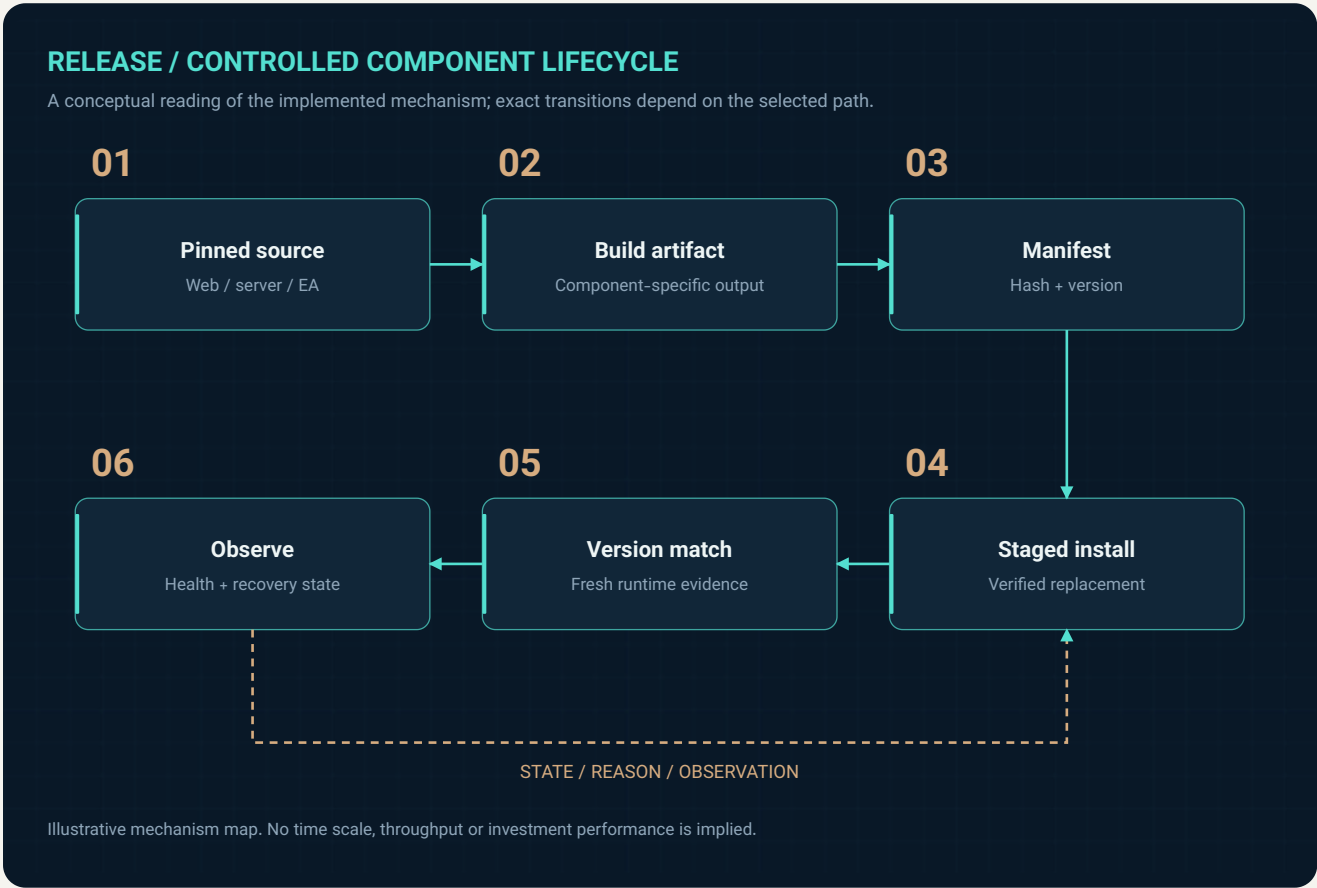
03 Administrative controls

Panels cover strategies, risk, hours, accounts, services, models, logs and permissions. Each control is governed by its authority scope and server-side validation.

04 / DEPLOYMENT AND RELEASE LIFECYCLE

Version control across the product

Backend, web, client and EA releases have separate contracts, with component versions and evidence preserved.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Backend

The release contract uses committed input, compatible cohorts, tests, immutable hashes, transactional swap and rollback. Trading readiness additionally requires multi-cycle health and valid data.

02 Web and client

Web releases use a dedicated publisher and a new version. EA/client binary changes belong to the client orchestrator; a web release must not replace them.

03 This document's baseline

This catalog uses pinned production-unified commit b90f26a. It contains web package 1.2.235 and EA source version 3.46; these do not prove versions installed across customer accounts.

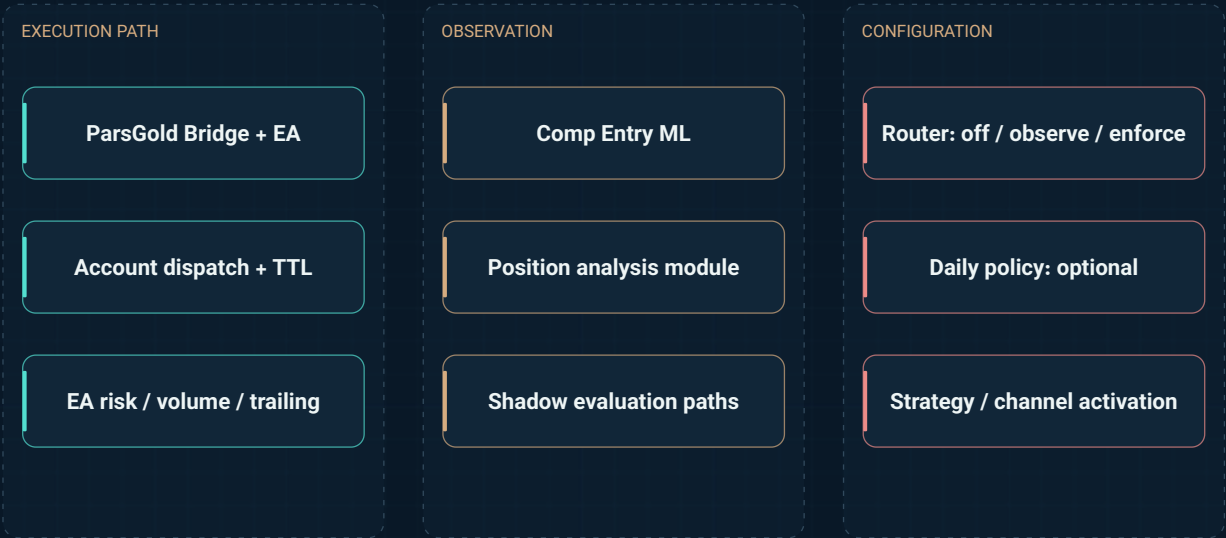
04 / CAPABILITY STATUS AND EVIDENCE

Implementation, authority and runtime state

Accurate product presentation distinguishes implementation, enabled configuration and measured performance.

EVIDENCE / DIFFERENT LEVELS OF AUTHORITY

Implementation, observation and enforcement must be distinguished in product evaluation.



Source presence is not proof of enabled production state. No audited return or latency claim is made.

Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 Implemented paths

The reviewed source contains the web app, APIs, data layer, analysis engines, USL, account queue, Bridge, EA and updater. Their presence does not establish that every component is enabled in one deployment.

02 Observation or optional enforcement

Comp Entry ML is observation-only; Position Management analysis is read-only; routing, hours and Daily Profit have modes or enforcement flags. Runtime configuration selects models and enabled rules.

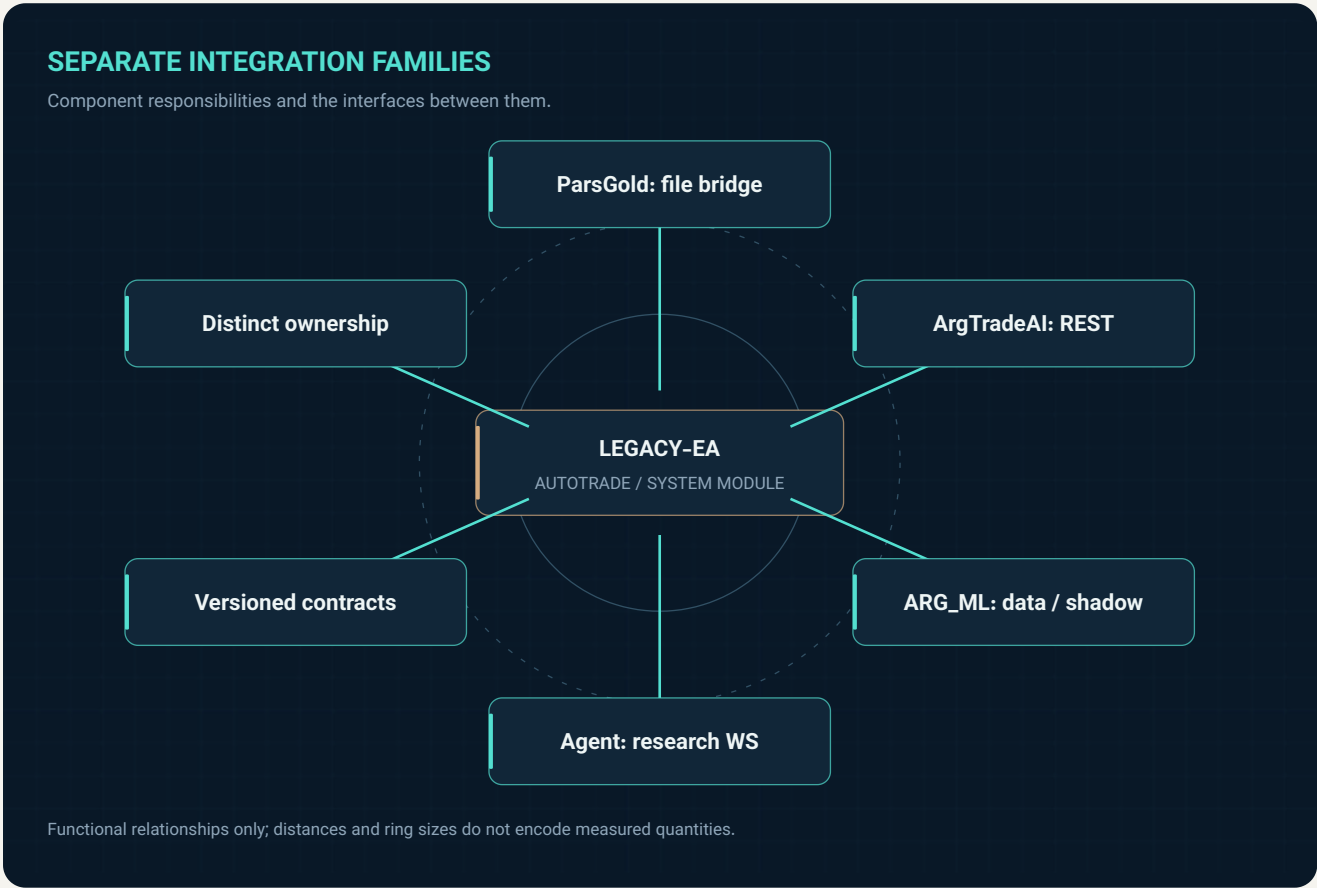
03 Screenshot evidence

Screenshots use the actual locally rendered interface with synthetic fixtures and blocked external communication. Surfaces without account or trained-model evidence are shown as empty states.

04 / ADDITIONAL EXPERT ADVISOR PATHS

Distinguish the additional EA paths

Multiple Expert Advisors in the repository do not imply simultaneous use or identical protocols.



Original diagram grounded in source mechanisms; placement does not encode measured scale or performance.

01 ArgTradeAI EA · 1.00

This source shows a legacy direct-HTTP path: terminal registration, heartbeat carrying account, positions and ticks, command reception and execution reporting. WebRequest runs directly inside MQL5.

02 ARG ML Utility EA · 2.00

The mql5 variant includes multi-timeframe data collection, regime detection, trade tagging and shadow trading. Chart-symbol detection and a status panel are present; simulated outcomes are distinct from live orders.

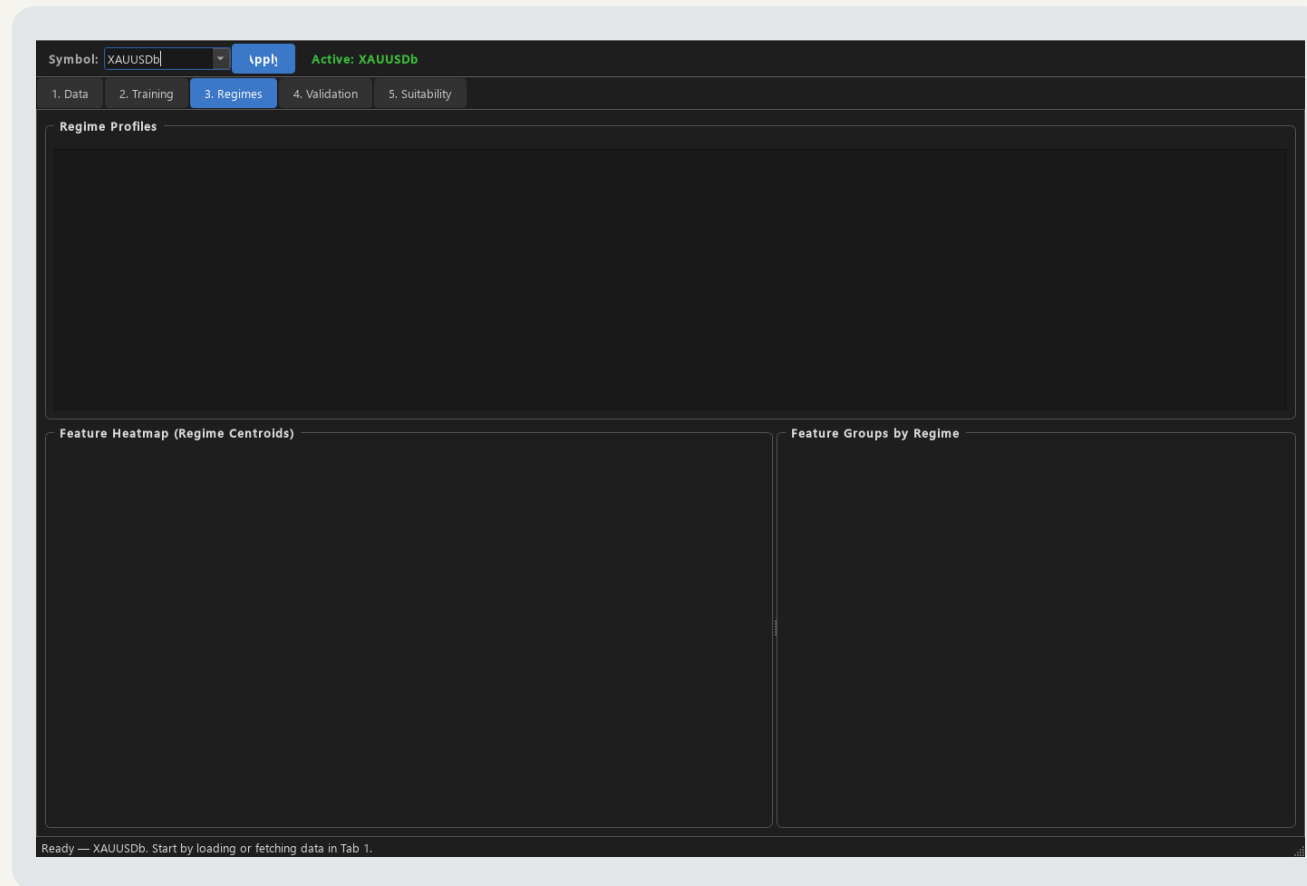
03 Selecting a deployment path

ParsGold places networking in the Bridge and execution in the EA; the legacy EA places HTTP inside the terminal. Configuration, licensing and releases must match the selected family. Installation or activation of these companion variants in customer environments was not verified.

04 / DESKTOP RESEARCH AND MODELLING TOOLS

Research tools behind the models

Research tools separate data preparation, training, validation and model export from order execution.



Actual desktop interface; synthetic state or no loaded model.

01 ARG Regime Classifier

The PyQt interface provides Data, Training, Regimes, Validation and Suitability tabs. Users can load data, review features, train a GMM and assess regimes and strategy suitability.

02 ARG Sharp Move Intelligence

Data, Detection, Patterns, HMM, Export and Validation tabs cover sharp-move analysis. Shared state includes M1 data, 15-dimensional features, a VQ clustering codebook and an HMM model.

03 What the screenshots establish

Screenshots show actual interfaces with no loaded dataset or trained model. MT5 data retrieval, training and model publication were not performed for this catalog; an Export control does not imply automatic backend activation.

NEXT / A PRACTICAL EVALUATION

From product introduction to operational evaluation

Required scope

Define roles, accounts, symbols, broker and selected strategies.

Delivery and readiness

Check feed freshness, account identity, TTL, reconnect and EA readiness.

Evidence assessment

Separate replay and shadow output from actual results; agree evaluation criteria before execution.

Component alignment

Align web, server, Bridge and EA versions with the time contract and authoritative configuration.

Demo-account execution

Verify order acceptance or rejection, volume, SL/TP, trailing and account/signal attribution.

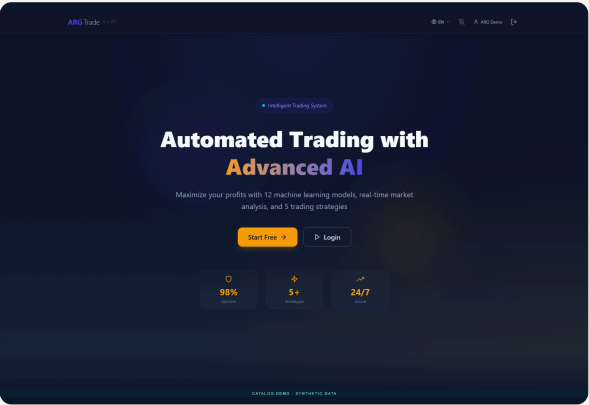
Handover and operation

Define monitoring, operations ownership, notifications, versioning and recovery for the target installation.

WEB 1.2.235	BACKEND 4.3.29	PARSGOLD EA SOURCE 3.46	BASELINE b90f26a
-------------	----------------	-------------------------	------------------

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

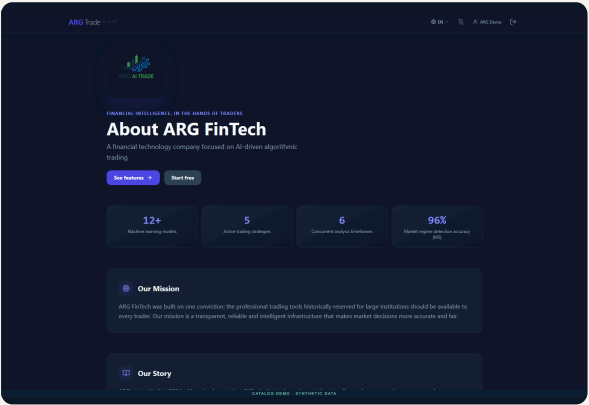


/landing

01 / WEB

Product landing

Product introduction and entry into the analytical workspace, linking capabilities, account access and public information.



/about

02 / WEB

About the platform

Introduces the product family and the roles of web, backend and EA. Embedded promotional assets are part of the product presentation and are not necessarily current runtime captures.

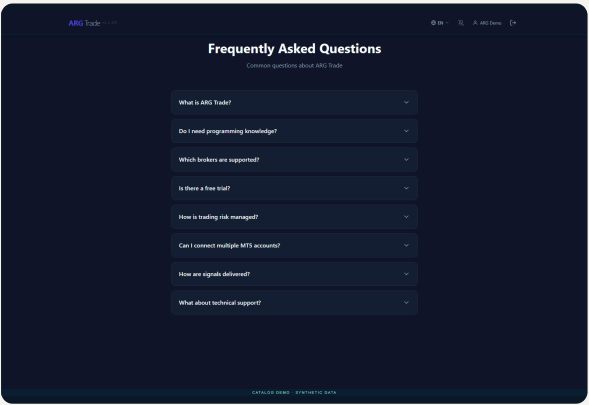


/features

03 / WEB

Feature overview

Feature presentation for analysis, regimes, strategy authoring and alerts. The architecture chapters clarify each capability's implementation status and authority.



/faq

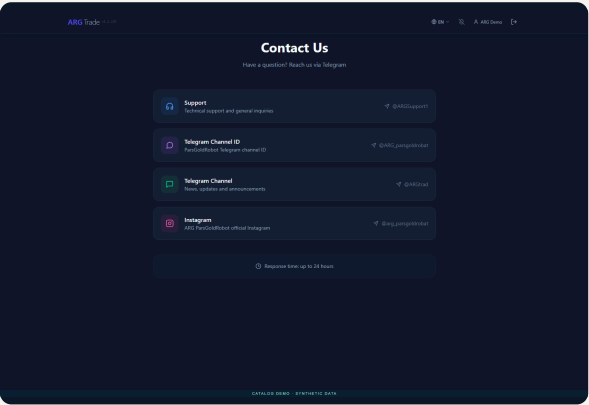
04 / WEB

Frequently asked questions

A support surface for onboarding and product-use questions. Help content does not constitute a trading-outcome guarantee.

Web interface visual index

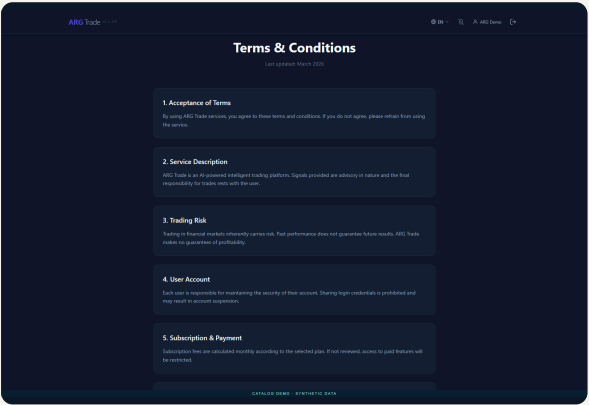
Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.



/contact

05 / WEB Contact and support

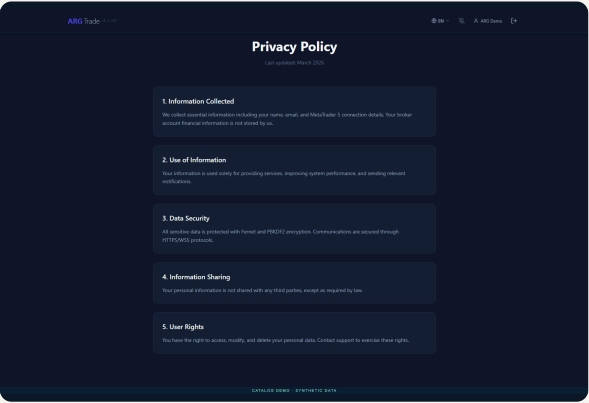
Public contact and support entry, separate from trading-account controls.



/terms

06 / WEB Terms of use

Terms-of-use interface. Customer-facing terms should be checked against the current contractual edition; this image documents the interface.



/privacy

07 / WEB Privacy information

Privacy and data-information page. Published policy text is separate from independent security-audit evidence.



/track-record

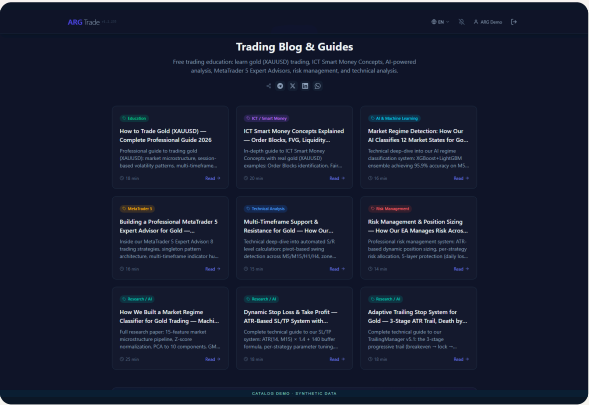
08 / WEB Track record

Public performance and track-record surface. The local edition does not load real account records or audited profitability evidence.

Actual application interface, captured locally with synthetic data.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

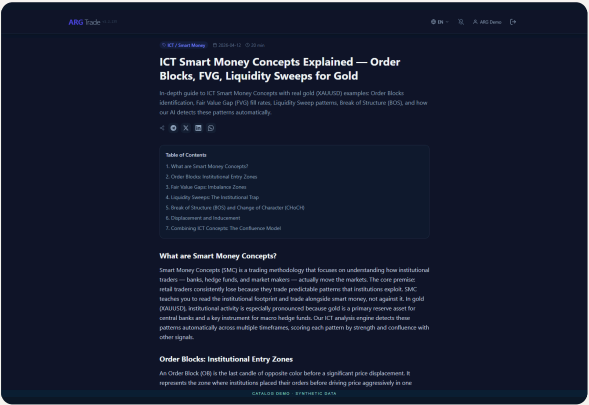


/blog

09 / WEB

Editorial library

Educational content covering analysis, regimes, risk concepts and platform use.

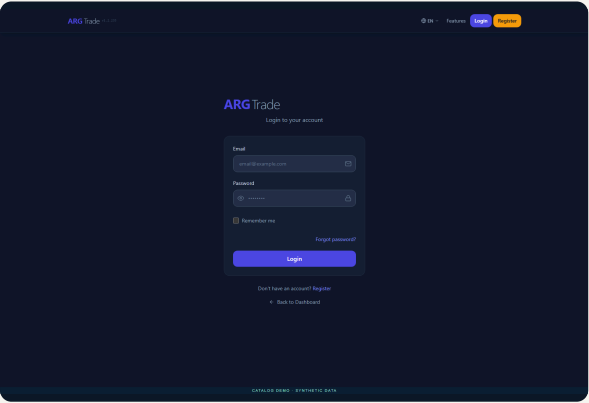


/blog/:slug

10 / WEB

Educational article

An ICT and Smart Money article from the application's own content. Its title, text and page structure come from the source application.

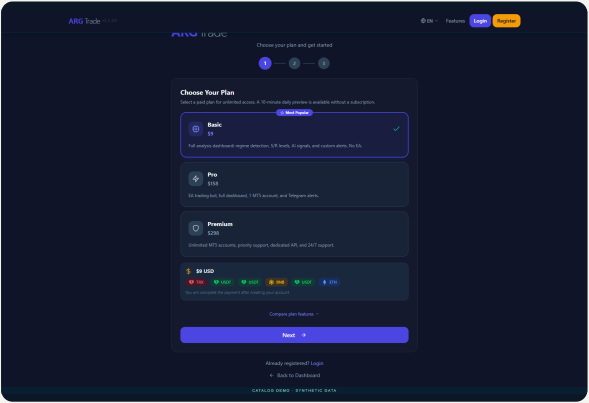


/login

11 / WEB

Account sign-in

The sign-in form establishes user identity. In the actual system, token and profile validation precede protected access.



/register

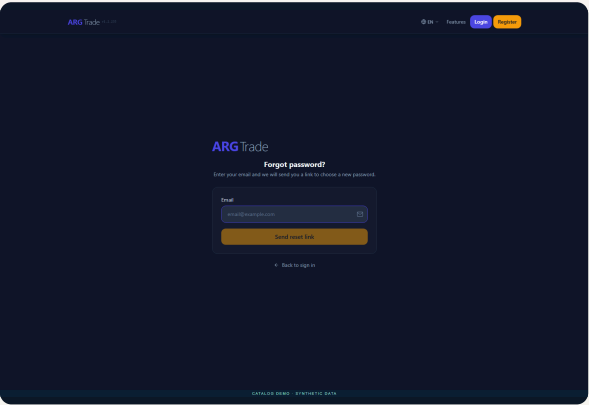
12 / WEB

Registration

Collects registration details, plan selection and acceptance of terms. No real registration was performed to produce this image.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

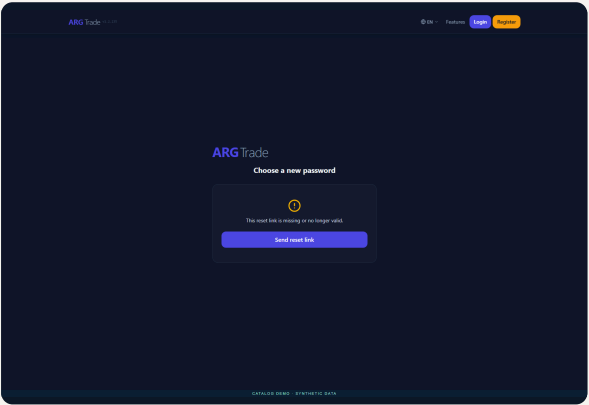


/forgot-password

13 / WEB

Password recovery

Password recovery starts from a dedicated request form and depends on server-side identity checks.

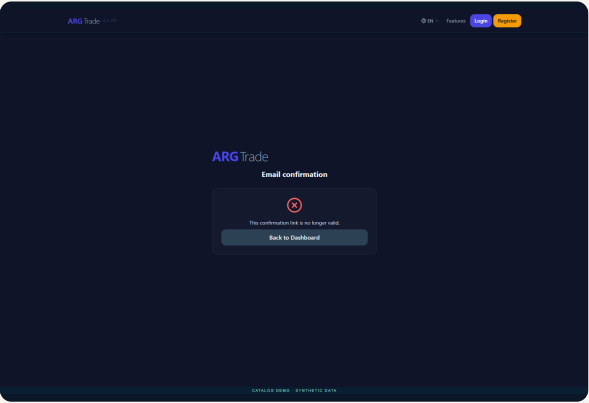


/reset-password

14 / WEB

Password reset and link validation

Shows a reset link without a valid token. Link validation is part of recovery; no successful reset state was fabricated.

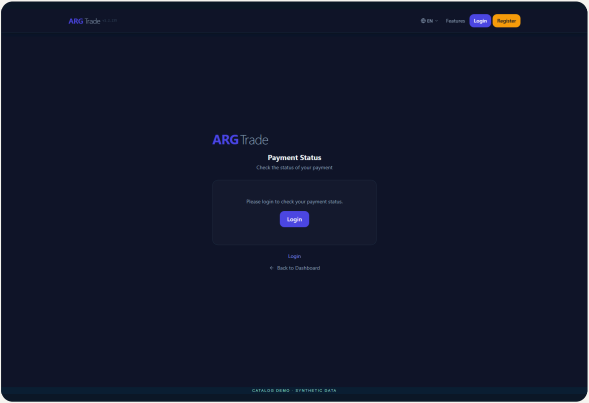


/verify-email

15 / WEB

Email verification

Email verification-link validation. The shown state belongs to a demonstration link, not a completed user verification.



/payment-status

16 / WEB

Payment status

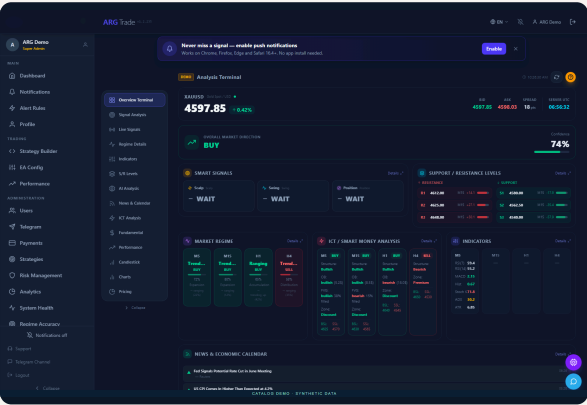
Subscription and payment-status tracking. The screenshot contains no real payment or transfer.

Actual application interface, captured locally with synthetic data.

APPENDIX A / 17–20 OF 61

Web interface visual index

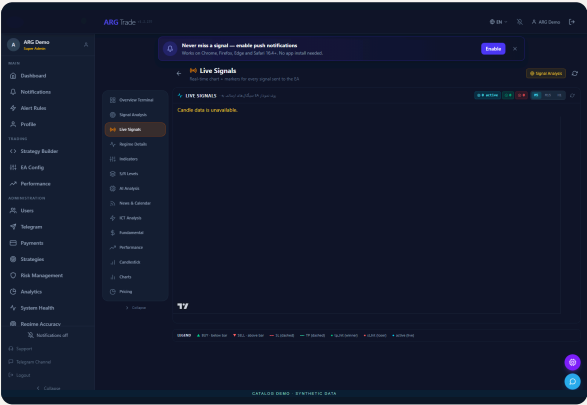
Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.



/dashboard/overview

17 / WEB
Unified analysis terminal

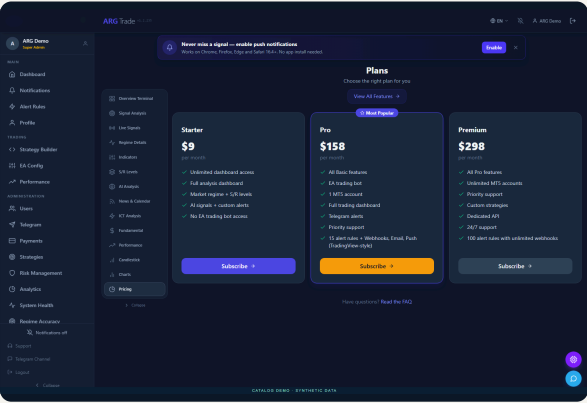
Combines quote, directional context, horizon signals, levels, regimes, ICT, indicators and news in one workspace for rapid market-context review.



/dashboard/signals

18 / WEB
Signal list

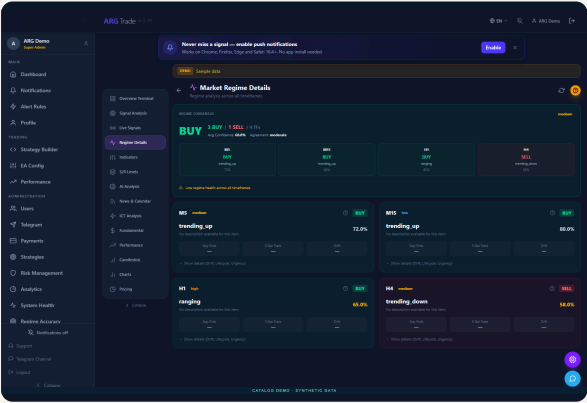
Lists signals and their state. Displayed direction must be interpreted alongside dispatch permission, expiry and execution outcome.



/dashboard/pricing

19 / WEB
Plans and access

Compares product access tiers. Prices inside the screenshot are not a commercial quotation; a sales proposal must use the current plan configuration.



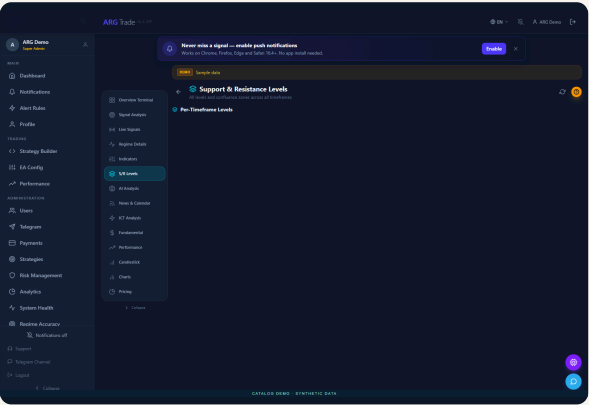
/dashboard/regime

20 / WEB
Market-regime detail

Shows regimes, direction, confidence and transition context across timeframes. Model confidence describes market-state inference, not a trade win probability.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

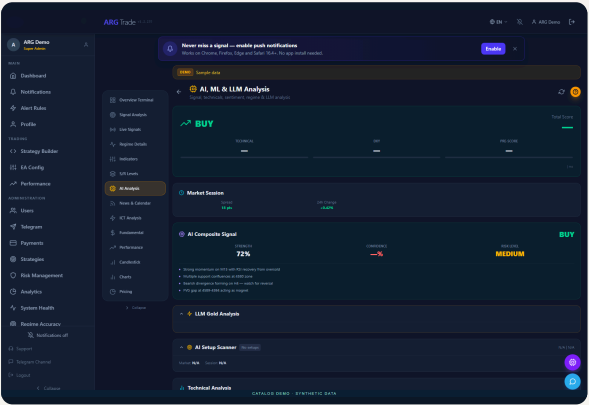


/dashboard/sr

21 / WEB

Support and resistance

Support/resistance levels and price distance help identify entry context, rejection and breakout. Final decisions still depend on signal lifecycle and policy.

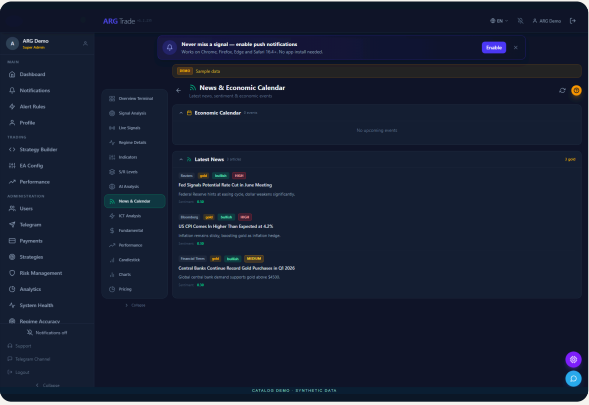


/dashboard/ai

22 / WEB

AI analysis

Presents market-grounded analysis and explanatory factors. User-facing interpretation is separate from order-execution ownership.

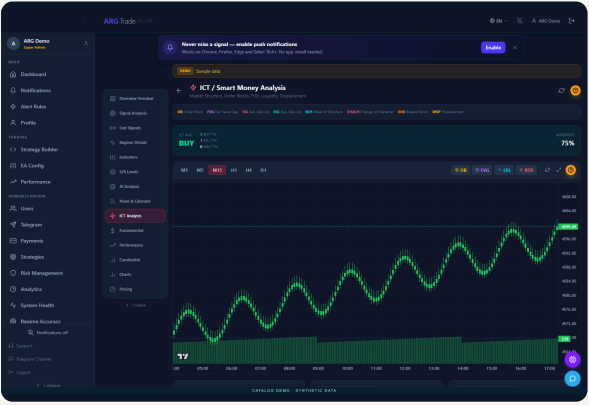


/dashboard/news

23 / WEB

News and economic calendar

Combines news, source, category, sentiment and calendar events. Screenshot content is fixture data, not a live news feed.



/dashboard/ict

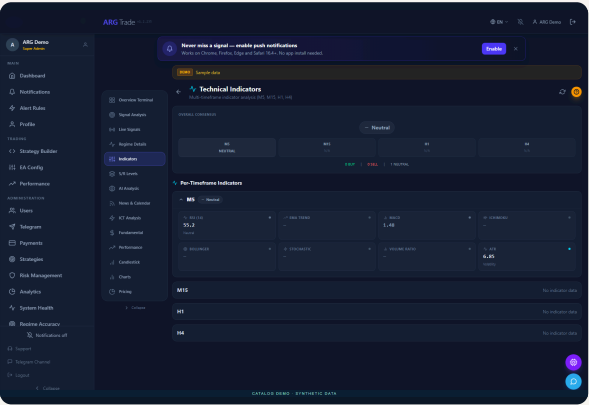
24 / WEB

ICT and liquidity analysis

Displays market structure, order blocks, FVGs and liquidity zones across timeframes to explain entry context and price location.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

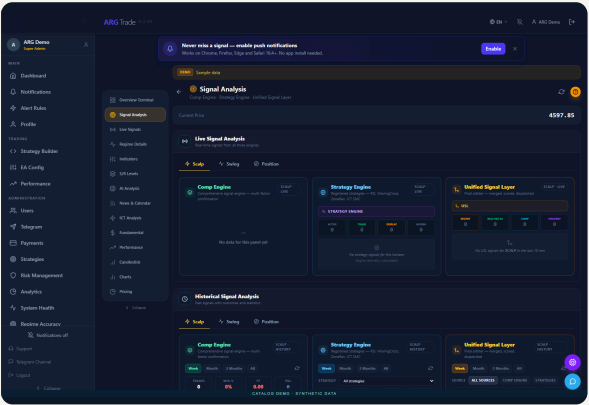


/dashboard/indicators

25 / WEB

Technical indicators

Indicator values and states across timeframes. The illustrated values are samples and were not used to execute trades.

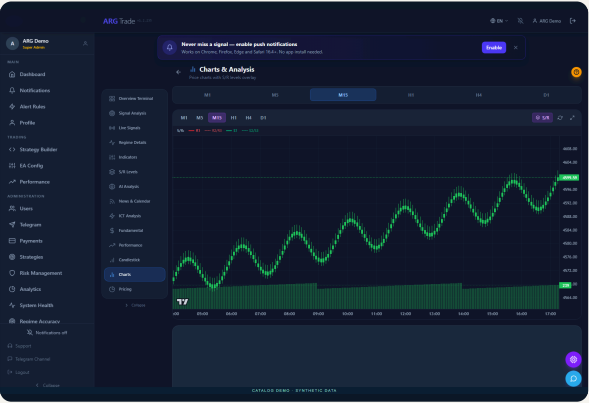


/dashboard/signal

26 / WEB

Signal analysis

Details signal factors, horizon and wait/proposed-direction state. Read this view together with the Comp and USL architecture chapters.

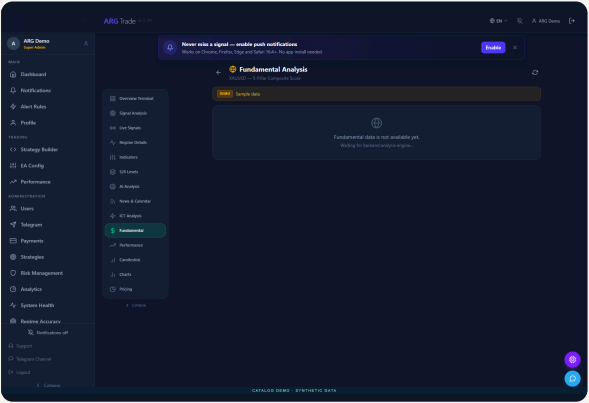


/dashboard/chart

27 / WEB

Price chart

Chart view for price movement and structure. Candle timestamps and timeframe must remain consistent with the feed contract and source.



/dashboard/fundamental

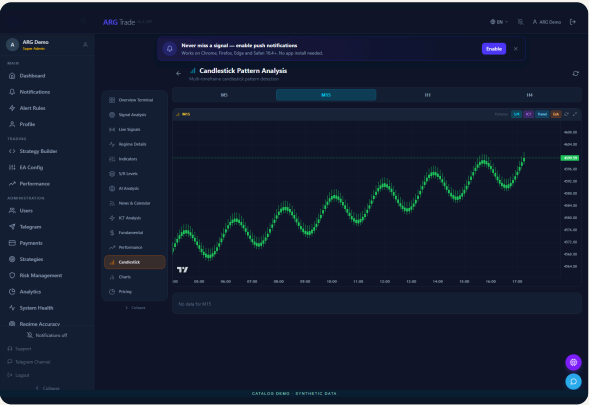
28 / WEB

Fundamental context

Shows fundamental and macro context relevant to analysis. Data and provider availability determine which content can be displayed.

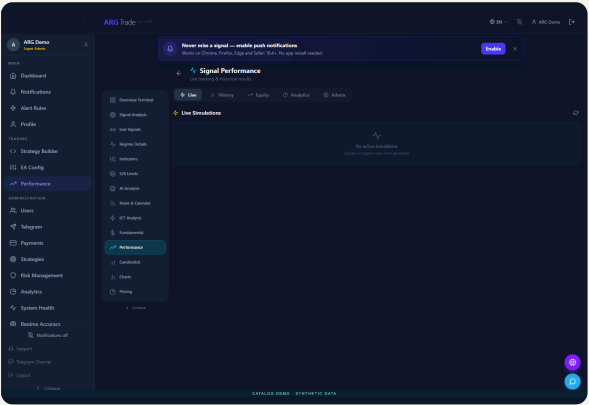
Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.



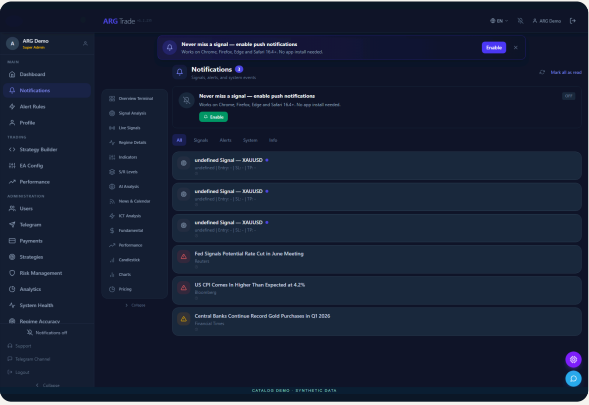
/dashboard/candles

29 / WEB
Candlestick patterns
Candlestick information and patterns in timeframe context. Patterns are analytical inputs, not standalone order authorization.



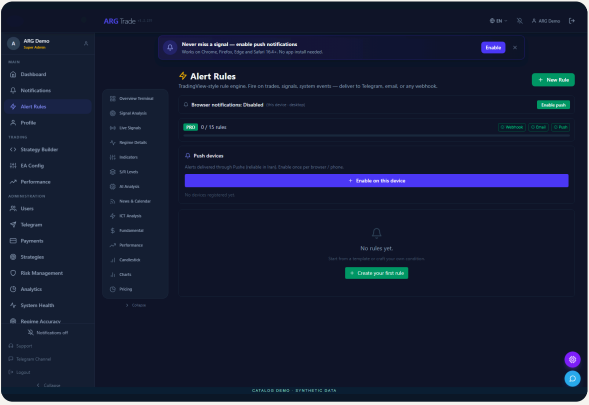
/dashboard/performance

30 / WEB
Performance analysis
History, statistics, breakdowns and performance curves. Figures require actual, shadow or simulation provenance; this view does not load a real account.



/dashboard/notifications

31 / WEB
Notification center
User notifications, delivery preferences and channel status. User alerts are distinct from the account-scoped trading queue and its acknowledgements.

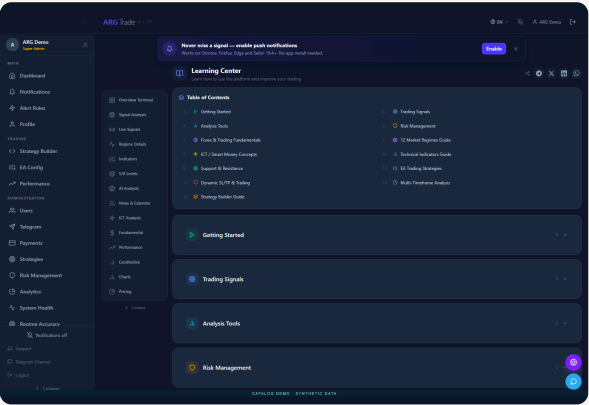


/dashboard/rules

32 / WEB
Alert rules
Defines alert conditions, messages, channels and plan limits, processed by the separate notification service.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

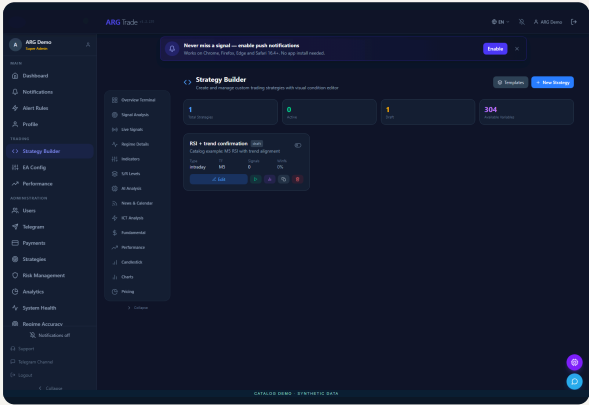


33 / WEB

Learning center

Explains analysis, market concepts, indicators, risk and tool use as part of the product experience.

/dashboard/learn

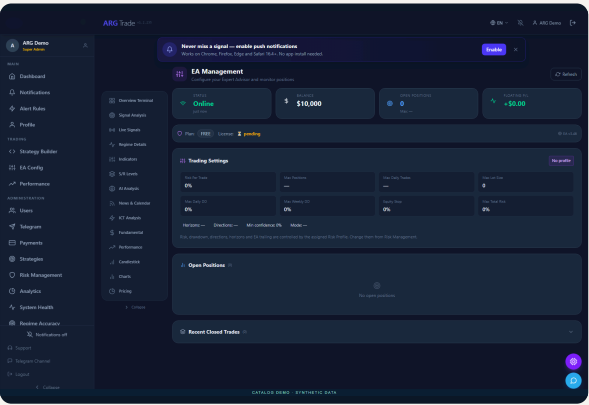


34 / WEB

Strategy Builder

Lists drafts and strategies with variables, editing, evaluation and backtesting. The inspected registry supplied 304 variables to the local interface.

/dashboard/strategy-builder



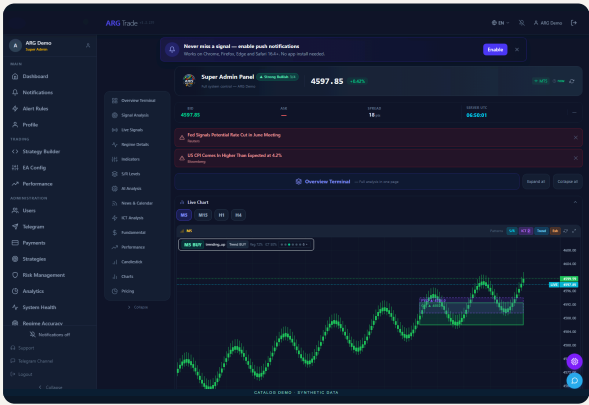
35 / WEB

Account EA management

Retrieves account state, runtime settings and EA positions through account-scoped APIs. Displayed settings were not sent to a real account.

/dashboard/ea

Actual application interface, captured locally with synthetic data.



36 / WEB

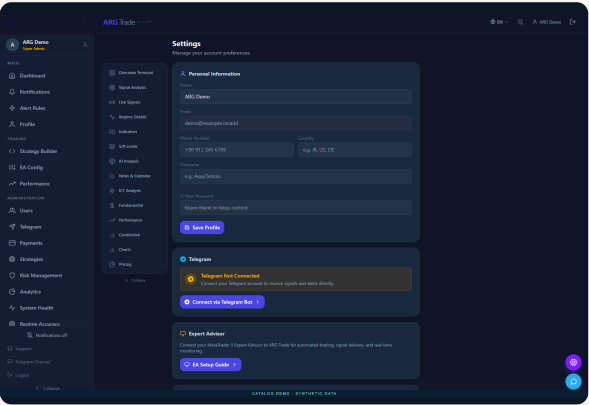
Role-aware dashboard

A unified dashboard with role- and preference-dependent content. Starter, Member and administration routes can share this component with different access.

/dashboard/starter

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

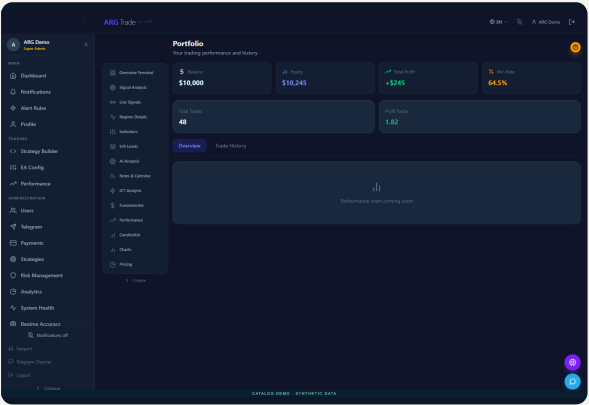


37 / WEB

Profile and user settings

Profile and user preferences are separate from trading-engine configuration. The displayed name and email belong to a synthetic catalog identity.

/dashboard/starter/settings

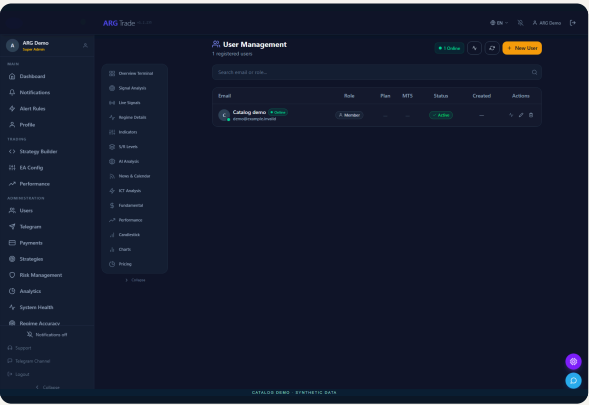


38 / WEB

Portfolio and account

User account, balance and position view. The screenshot documents page structure without exposing a customer's balance or history.

/dashboard/member/portfolio



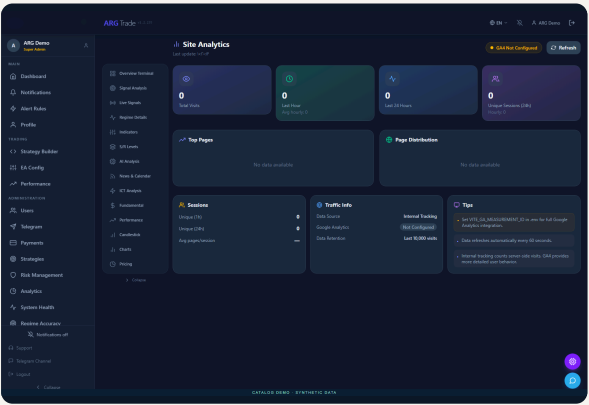
39 / WEB

User administration

Manages user identity, roles and state. A web account differs from a trading login; ownership linkage must be explicit.

/dashboard/admin/users

Actual application interface, captured locally with synthetic data.



40 / WEB

Usage analytics

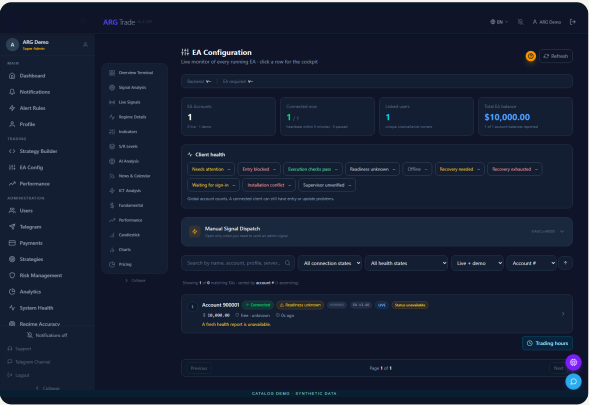
Application-visit and usage analytics, distinct from trading-account profit and loss.

/dashboard/admin/analytics

APPENDIX A / 41–44 OF 61

Web interface visual index

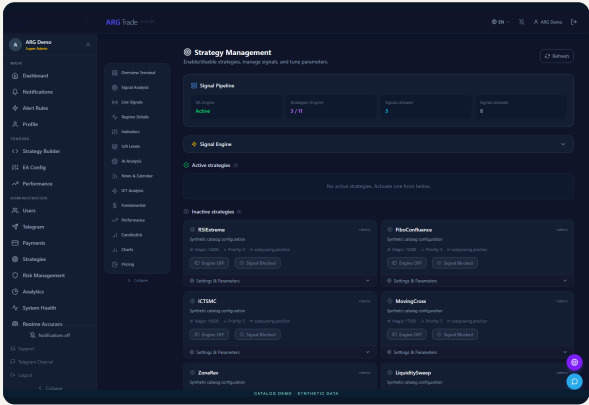
Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.



41 / WEB
EA fleet and configuration

Filters accounts by connection, health, version and state. Separating connected status from execution readiness reveals clients that communicate but cannot trade.

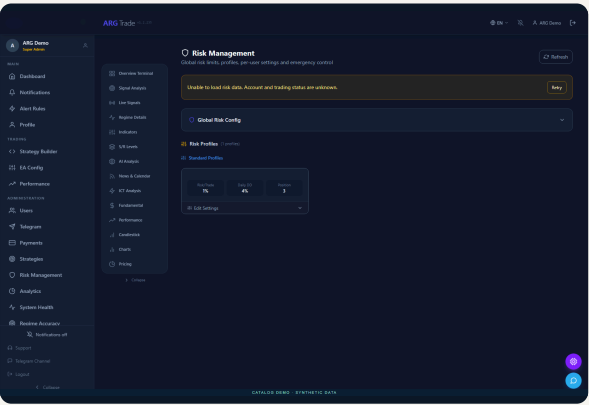
/dashboard/admin/ea-config



42 / WEB
Strategy administration

Controls engine participation, signal permission and strategy parameters. Horizons and magic identifiers connect policy to execution outcomes.

/dashboard/admin/strategies

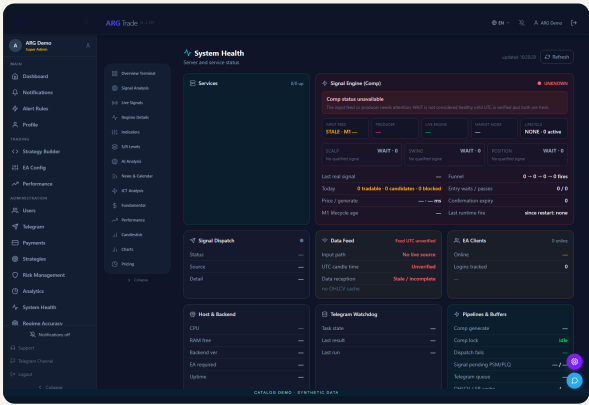


43 / WEB
Risk management

Global settings, profiles and account states, maintaining the boundary between account exposure and strategy policy.

/dashboard/admin/risk

Actual application interface, captured locally with synthetic data.



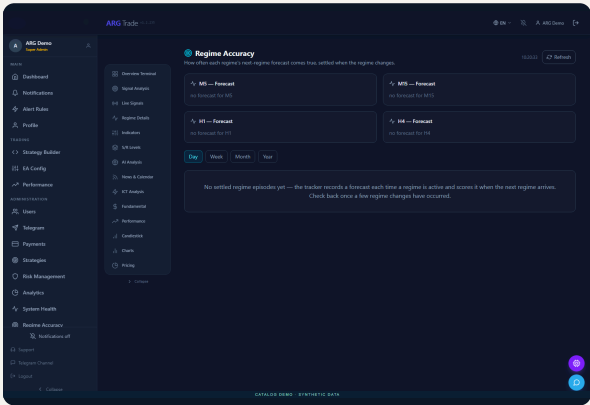
44 / WEB
System health

Component and data/analysis health for diagnosis. The catalog does not connect to production, so these indicators are not a public-service health claim.

/dashboard/admin/system-health

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

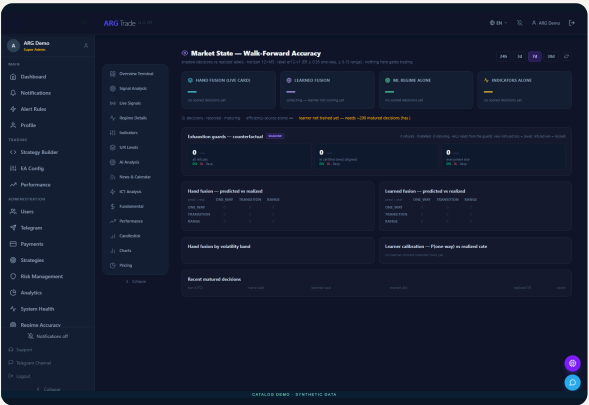


45 / WEB

Regime evaluation

Evaluates regime classification quality. A metric without its time window, sample and labeling method is not commercial performance evidence.

/dashboard/admin/regime-accuracy

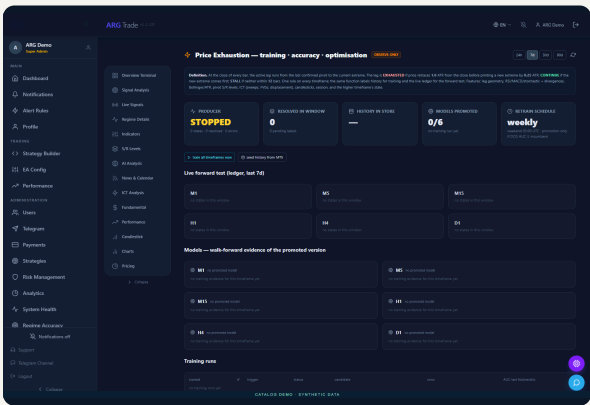


46 / WEB

Market-state evaluation

Assesses the market-state observer and router, comparing observed suggestions with execution behavior before enforcement decisions.

/dashboard/admin/market-state



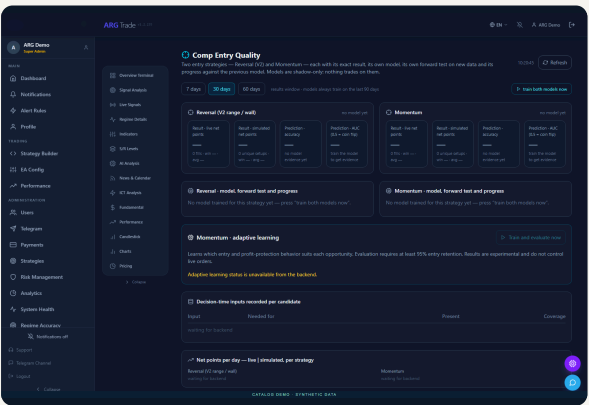
47 / WEB

Exhaustion tracking

Collects exhaustion state, models, training and reports. Model outputs must be interpreted with their mode and consuming path.

/dashboard/admin/exhaustion

Actual application interface, captured locally with synthetic data.



48 / WEB

Comp entry quality

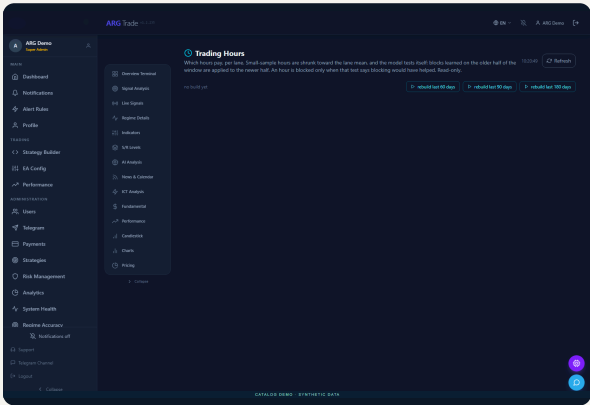
Shows features, models and candidate-entry evaluation. The reference ML runtime is observe-only and does not independently block entries.

/dashboard/admin/comp-entry

APPENDIX A / 49–52 OF 61

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

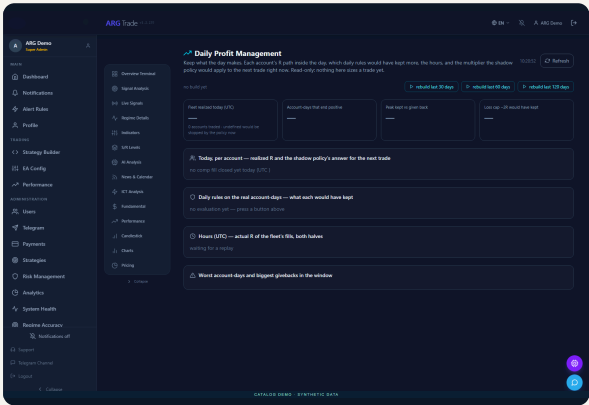


/dashboard/admin/trading-hours

49 / WEB

Trading-hours analysis

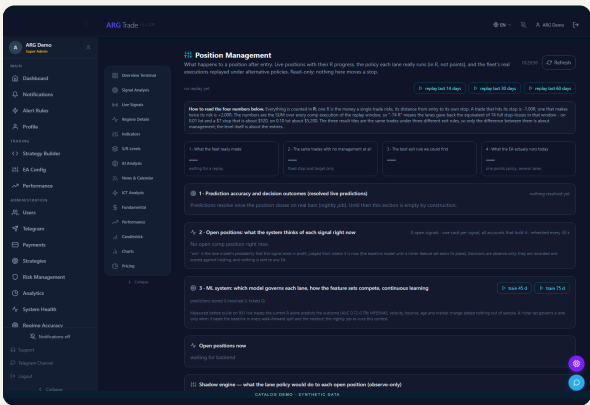
UTC-hour analysis for reversal and momentum with sample and temporal checks; learning and policy enforcement are separate.



50 / WEB

Daily profit and loss policy

Daily R path, giveback and policy-effect view. Actual account impact depends on enforcement flags and account scope.

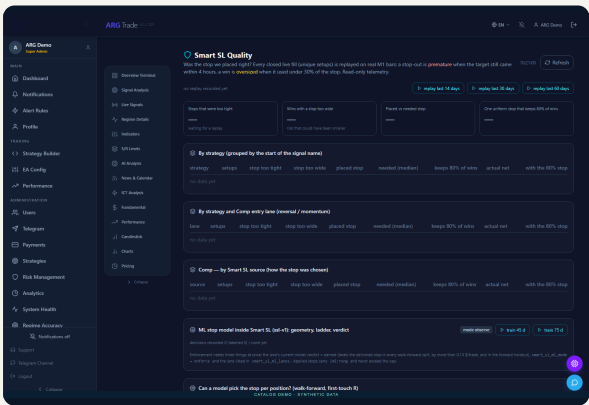


/dashboard/admin/positions

51 / WEB

Position-management analysis

Live positions, R-based policies and exit experiments. The analytical surface is read-only and distinct from direct execution controls.



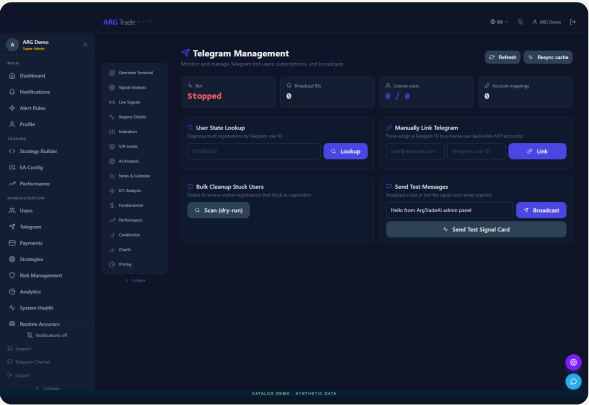
52 / WEB

Smart-stop quality

Evaluates stop distances, outcomes and related models, separately from initial geometry selection and EA execution.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

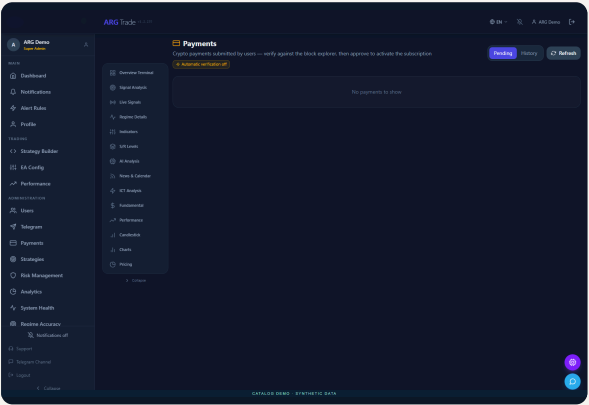


/dashboard/admin/telegram

53 / WEB

Telegram administration

Telegram status and configuration in the operator interface. No real messages were sent for the catalog.

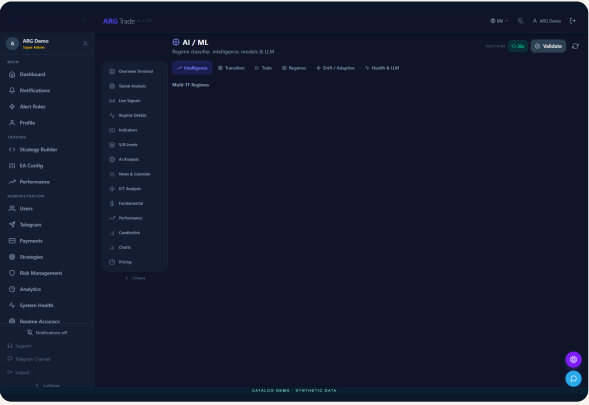


/dashboard/admin/payments

54 / WEB

Payment administration

Administrative payment and subscription status. Real customer financial data is not used in the local edition.

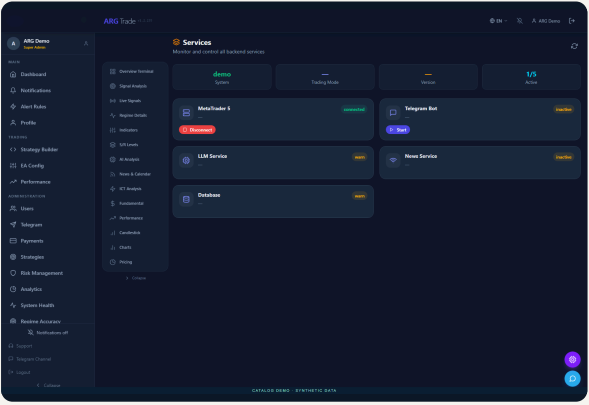


/dashboard/super/ai-ml

55 / WEB

AI and ML administration

Model and AI-health administration. Training and health controls do not imply that a training job ran during capture.



/dashboard/super/services

56 / WEB

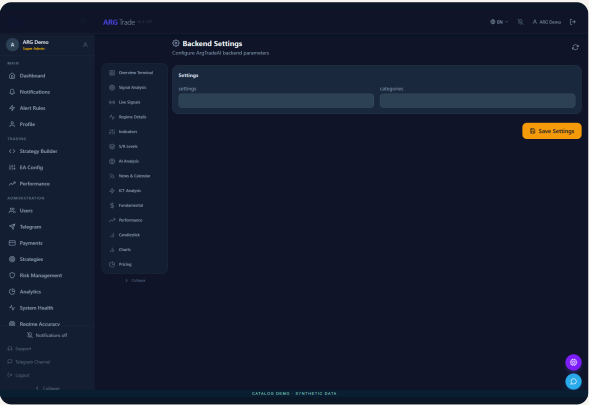
Services

Service-state view for operations. No real service was started, stopped or controlled in this demonstration.

APPENDIX A / 57–60 OF 61

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.

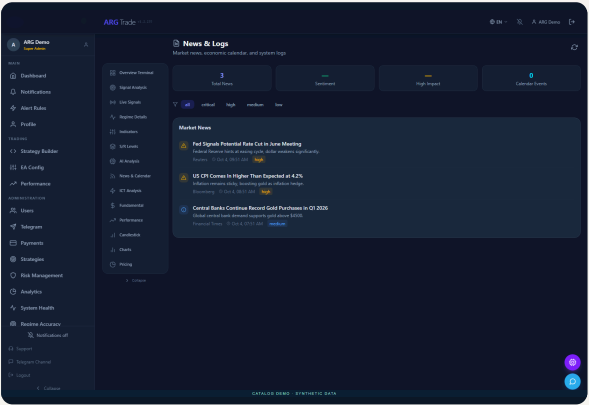


/dashboard/super/settings

57 / WEB

System settings

Super-administration settings and configuration categories. Real secrets and credentials are excluded from the demonstration.

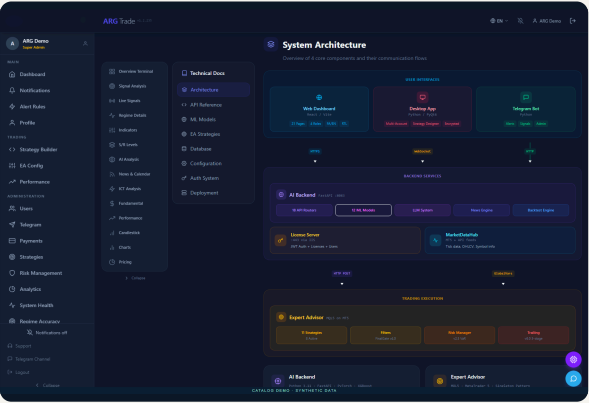


/dashboard/super/logs

58 / WEB

Logs and diagnostics

Event logs and diagnostic inspection. Catalog log views contain no sensitive records or real customer identifiers.

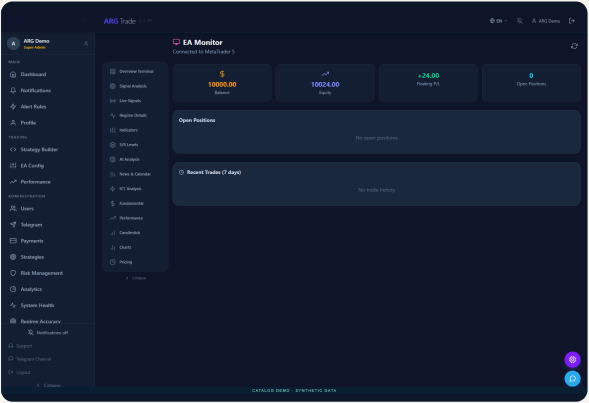


/dashboard/super/docs

59 / WEB

In-application documentation

Provides documentation, API information and component explanations inside the application. Descriptions must remain aligned with source and deployment versions.



/dashboard/super/ea-monitor

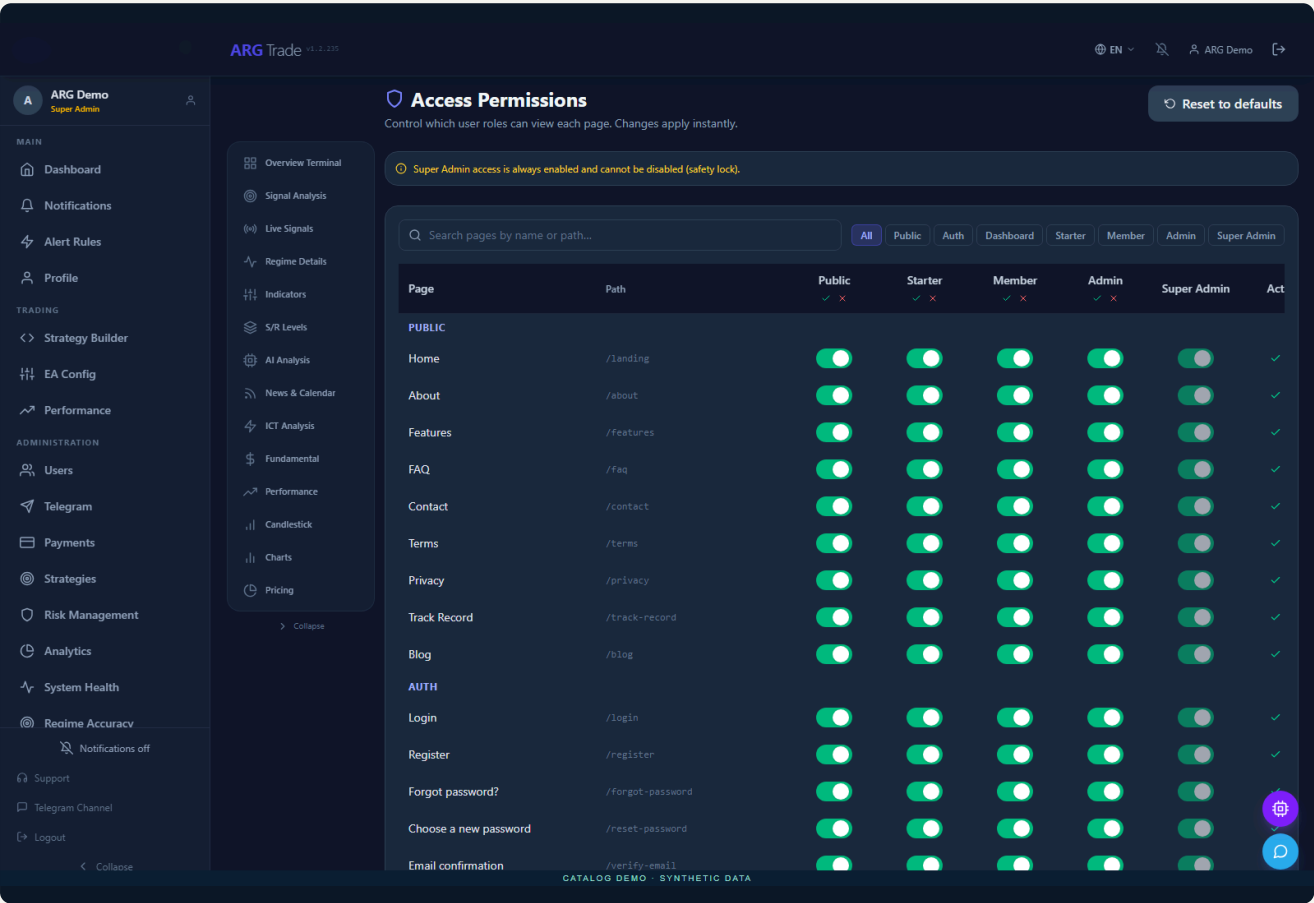
60 / WEB

MetaTrader monitor

Shows MT5 connectivity, account figures, positions and history. Screenshot figures are synthetic and do not evidence real trades.

Web interface visual index

Complete page index; key views appear at larger scale in the product chapters. Click an image in the HTML edition.



/dashboard/super/permissions

Actual application interface, captured locally with synthetic data.

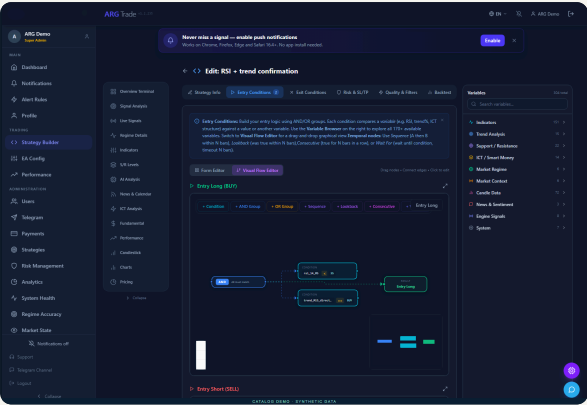
61 / WEB

Page-permission matrix

Defines role access to page groups. Server-side API authorization remains independent of whether a menu item is visible.

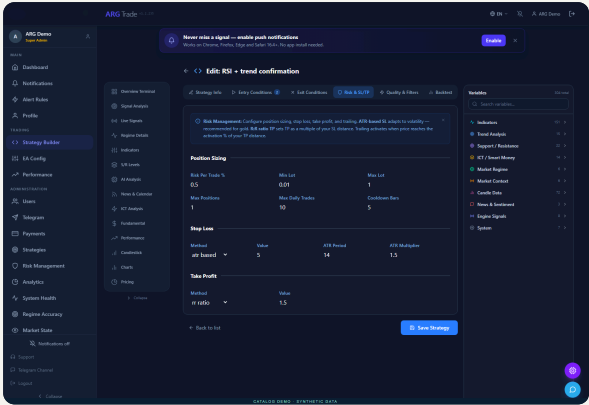
APPENDIX B / INTERACTION DETAILS

Product interaction details



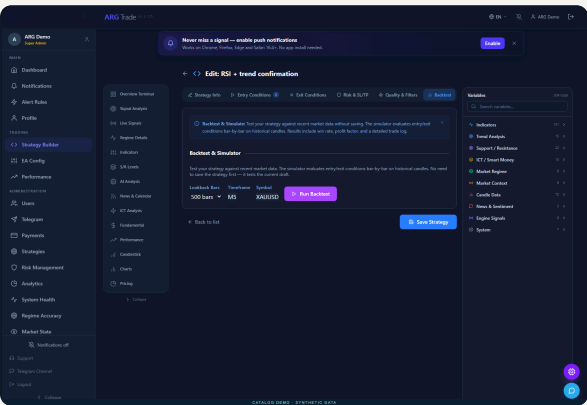
Visual flow editor

Actual React Flow condition nodes and connections. The illustrative draft was neither saved nor activated.



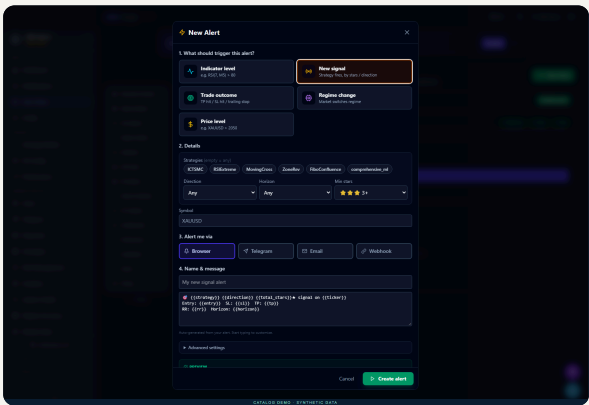
Builder risk and levels

Draft risk budget, volume, stop and target settings; final execution policy must remain aligned with the central layer.



Authored-strategy test

Draft test configuration. No trading simulation was executed during this catalog session.



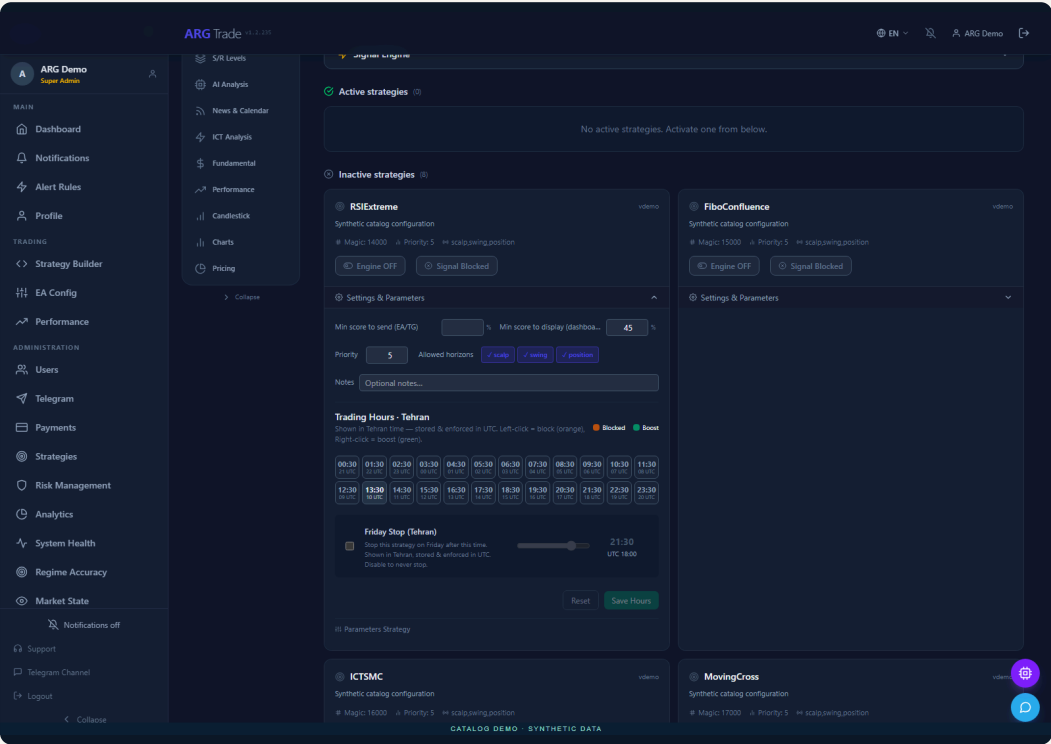
Alert-rule editor

Condition, message and channel authoring; no real notification was sent.

Actual application interface, captured locally with synthetic data.

APPENDIX B / INTERACTION DETAILS

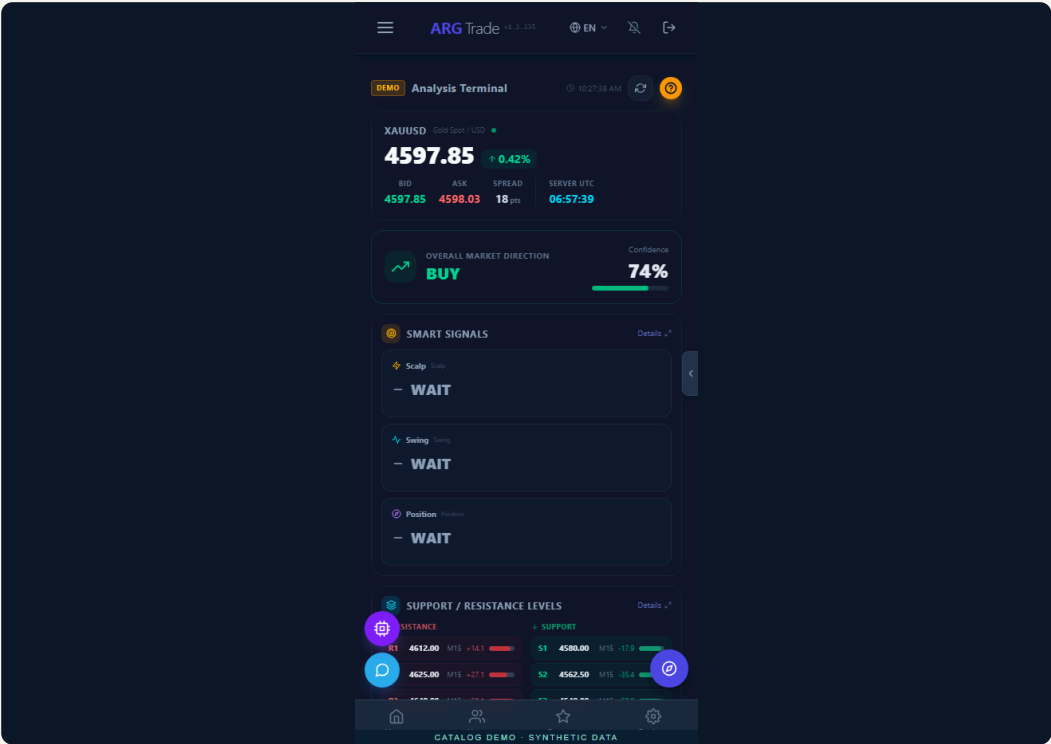
Product interaction details



Strategy parameters

Expanded strategy settings in the original panel. Values are catalog fixtures.

Actual application interface, captured locally with synthetic data.



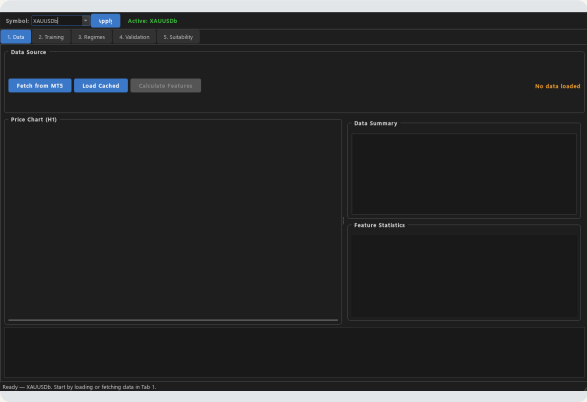
Mobile overview

Responsive analysis-terminal layout at a 430-pixel viewport.

APPENDIX C / DESKTOP TOOLS

Client and research-tool views

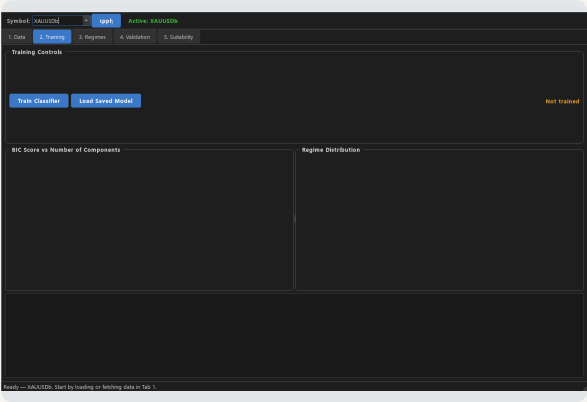
ParsGold uses a synthetic account, disconnected with trading off; research tools have no loaded dataset or trained model.



ArgRegimeClassifier

1. Data

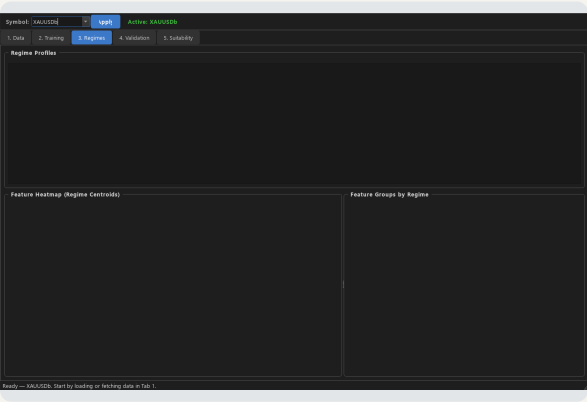
[ArgRegimeClassifier/ui/main_window.py](#)



ArgRegimeClassifier

2. Training

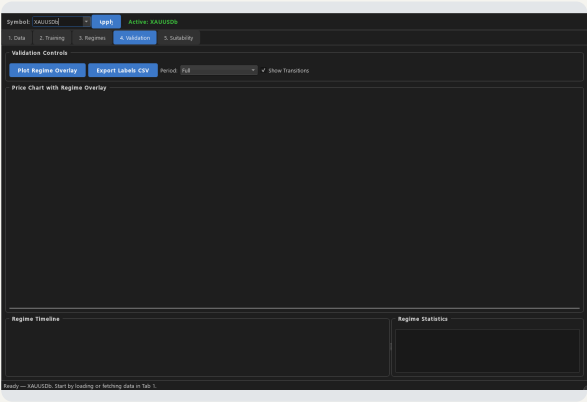
[ArgRegimeClassifier/ui/main_window.py](#)



ArgRegimeClassifier

3. Regimes

[ArgRegimeClassifier/ui/main_window.py](#)



ArgRegimeClassifier

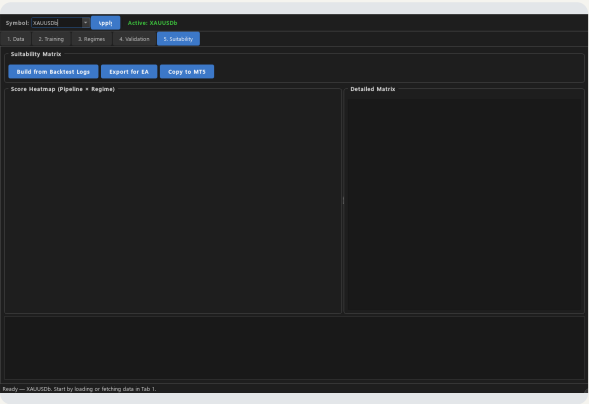
4. Validation

[ArgRegimeClassifier/ui/main_window.py](#)

APPENDIX C / DESKTOP TOOLS

Client and research-tool views

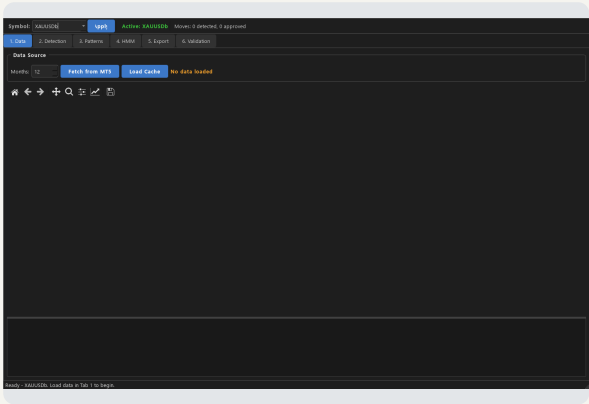
ParsGold uses a synthetic account, disconnected with trading off; research tools have no loaded dataset or trained model.



ArgRegimeClassifier

5. Suitability

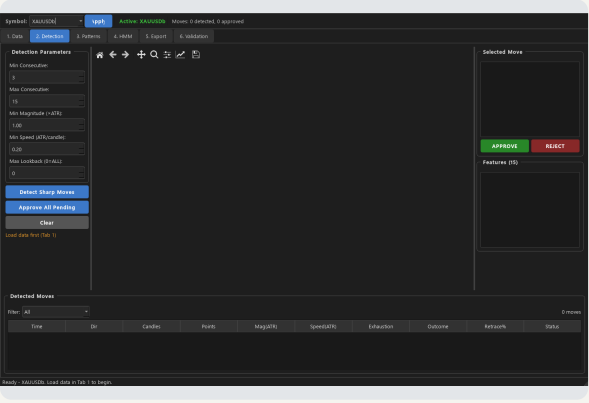
[ArgRegimeClassifier/ui/main_window.py](#)



ArgSharpMoveIntelligence

1. Data

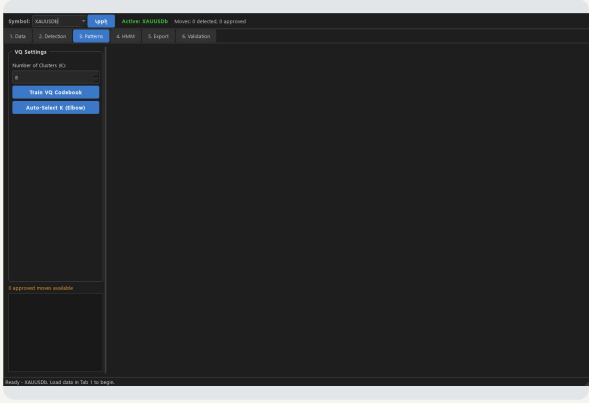
[ArgSharpMoveIntelligence/ui/main_window.py](#)



ArgSharpMoveIntelligence

2. Detection

[ArgSharpMoveIntelligence/ui/main_window.py](#)



ArgSharpMoveIntelligence

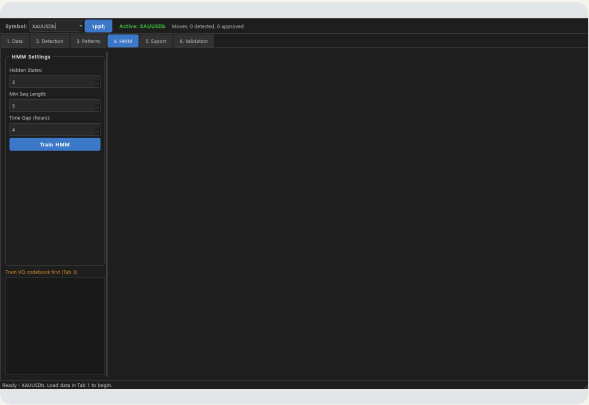
3. Patterns

[ArgSharpMoveIntelligence/ui/main_window.py](#)

APPENDIX C / DESKTOP TOOLS

Client and research-tool views

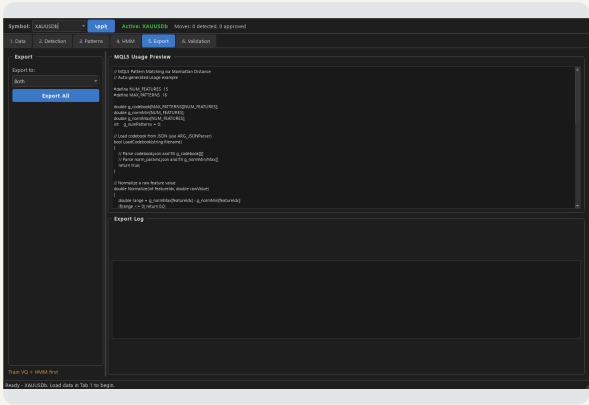
ParsGold uses a synthetic account, disconnected with trading off; research tools have no loaded dataset or trained model.



ArgSharpMoveIntelligence

4. HMM

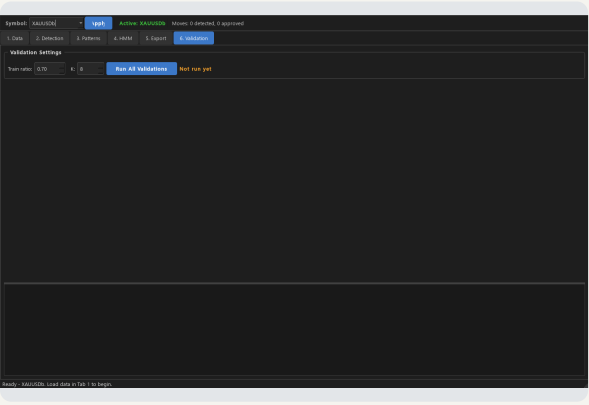
ArgSharpMoveIntelligence/ui/main_window.py



ArgSharpMoveIntelligence

5. Export

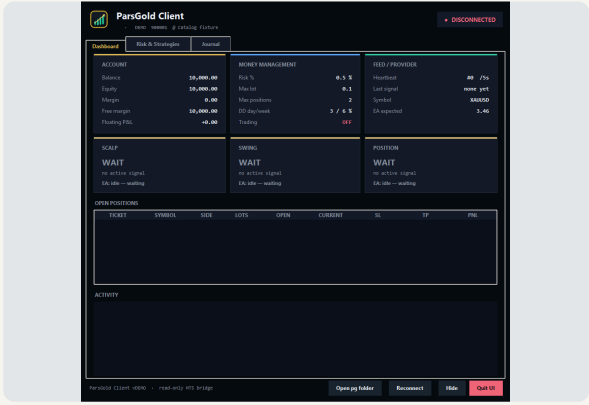
ArgSharpMoveIntelligence/ui/main_window.py



ArgSharpMoveIntelligence

6. Validation

ArgSharpMoveIntelligence/ui/main_window.py



ParsGold Client

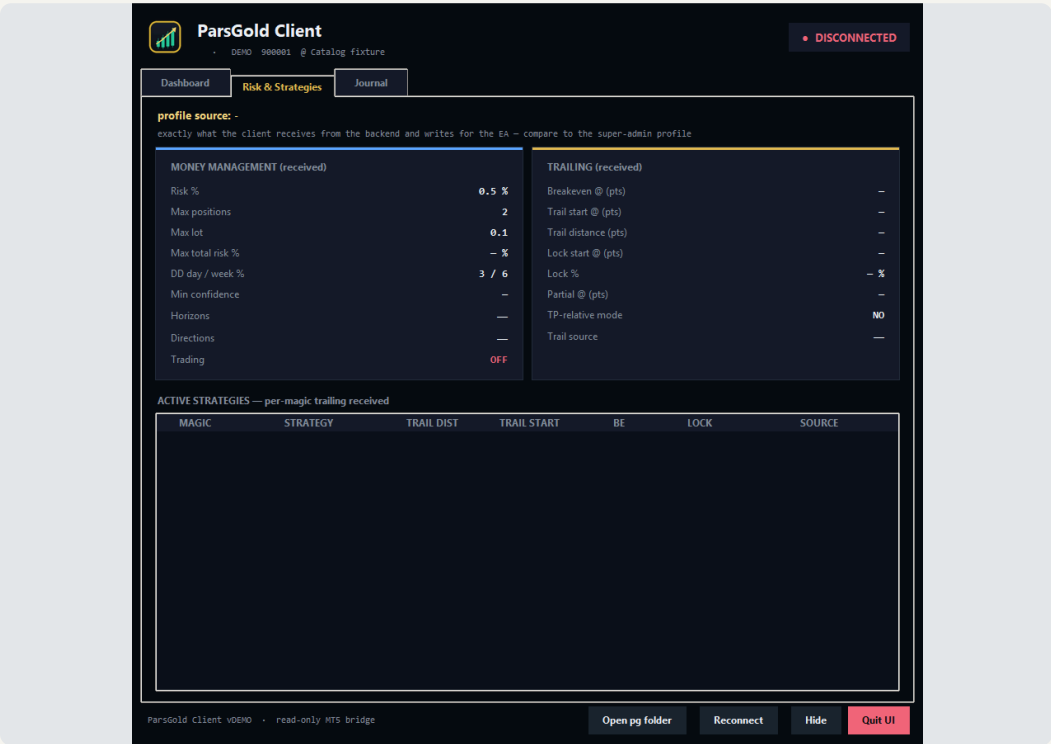
Dashboard

ParsGold_PyBridge/pg_ui.py

APPENDIX C / DESKTOP TOOLS

Client and research-tool views

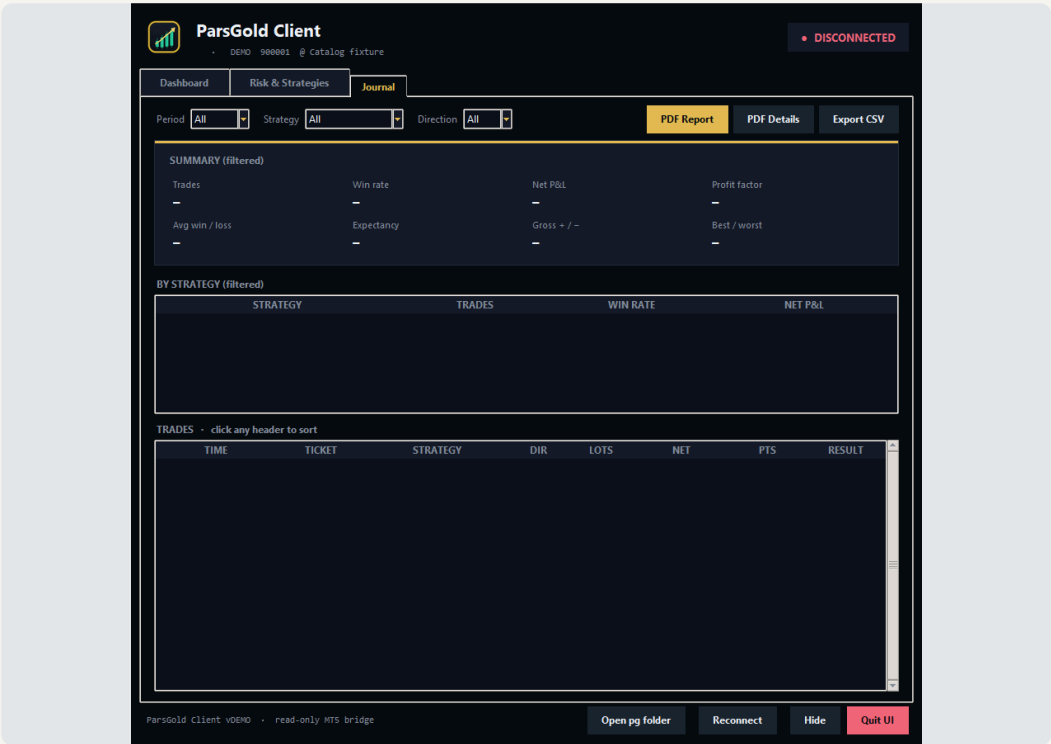
ParsGold uses a synthetic account, disconnected with trading off; research tools have no loaded dataset or trained model.



ParsGold Client

Risk & Strategies

ParsGold_PyBridge/pg_ui.py



ParsGold Client

Journal

ParsGold_PyBridge/pg_ui.py

APPENDIX D / ROUTE MAP

Route coverage and image references

Routes sharing a component refer to one image. A wildcard fallback is not a separate page.

Route	Component	Image page
/landing	LandingPage	47
/about	AboutPage	47
/features	FeaturesPage	47
/faq	FAQPage	47
/contact	ContactPage	48
/terms	TermsPage	48
/privacy	PrivacyPage	48
/track-record	TrackRecordPage	48
/blog	BlogPage	49
/blog/:slug	BlogPostPage	49
/login	LoginPage	49
/register	RegisterPage	49
/forgot-password	ForgotPasswordPage	50
/reset-password	ResetPasswordPage	50
/verify-email	VerifyEmailPage	50
/payment-status	PaymentStatusPage	50
/dashboard/overview	OverviewPage	51
/dashboard/signals	SignalsPage	51
/dashboard/pricing	PricingPage	51
/dashboard/regime	RegimeDetailPage	51
/dashboard/sr	SRDetailPage	52
/dashboard/ai	AIDetailPage	52
/dashboard/news	NewsDetailPage	52
/dashboard/ict	ICTDetailPage	52

APPENDIX D / ROUTE MAP

Route coverage and image references

Routes sharing a component refer to one image. A wildcard fallback is not a separate page.

Route	Component	Image page
/dashboard/indicators	IndicatorDetailPage	53
/dashboard/signal	SignalDetailPage	53
/dashboard/chart	ChartDetailPage	53
/dashboard/fundamental	FundamentalDetailPage	53
/dashboard/candles	CandlestickDetailPage	54
/dashboard/performance	PerformancePage	54
/dashboard/notifications	NotificationsPage	54
/dashboard/rules	RulesPage	54
/dashboard/learn	LearnPage	55
/dashboard/strategy-builder	StrategyBuilderPage	55
/dashboard/ea	EAManagePage	55
/dashboard/starter	Dashboard	55
/dashboard/starter/signals	SignalsPage	51
/dashboard/starter/notifications	NotificationsPage	54
/dashboard/starter/settings	SettingsPage	56
/dashboard/member	Dashboard	55
/dashboard/member/signals	SignalsPage	51
/dashboard/member/portfolio	PortfolioPage	56
/dashboard/member/notifications	NotificationsPage	54
/dashboard/member/settings	SettingsPage	56
/dashboard/admin	Dashboard	55
/dashboard/admin/users	UsersPage	56
/dashboard/admin/analytics	AnalyticsPage	56
/dashboard/admin/ea-config	EAConfigPage	57

APPENDIX D / ROUTE MAP

Route coverage and image references

Routes sharing a component refer to one image. A wildcard fallback is not a separate page.

Route	Component	Image page
/dashboard/admin/strategies	StrategiesPage	57
/dashboard/admin/risk	RiskManagementPage	57
/dashboard/admin/system-health	SystemHealthPage	57
/dashboard/admin/regime-accuracy	RegimeAccuracyPage	58
/dashboard/admin/market-state	MarketStateAccuracyPage	58
/dashboard/admin/exhaustion	ExhaustionTrackingPage	58
/dashboard/admin/comp-entry	CompEntryQualityPage	58
/dashboard/admin/trading-hours	TradingHoursPage	59
/dashboard/admin/daily-profit	DailyProfitPage	59
/dashboard/admin/positions	PositionManagementPage	59
/dashboard/admin/smart-sl	SmartSLQualityPage	59
/dashboard/admin/settings	SettingsPage	56
/dashboard/admin/telegram	TelegramPage	60
/dashboard/admin/payments	PaymentsPage	60
/dashboard/super	Dashboard	55
/dashboard/super/users	UsersPage	56
/dashboard/super/telegram	TelegramPage	60
/dashboard/super/payments	PaymentsPage	60
/dashboard/super/ai-ml	AIMLPage	60
/dashboard/super/services	ServicesPage	60
/dashboard/super/settings	SuperSettingsPage	61
/dashboard/super/logs	LogsPage	61
/dashboard/super/analytics	AnalyticsPage	56
/dashboard/super/docs	DocsPage	61

APPENDIX D / ROUTE MAP

Route coverage and image references

Routes sharing a component refer to one image. A wildcard fallback is not a separate page.

Route	Component	Image page
/dashboard/super/ea-config	EAConfigPage	57
/dashboard/super/ea-monitor	EAMonitorPage	61
/dashboard/super/strategies	StrategiesPage	57
/dashboard/super/risk	RiskManagementPage	57
/dashboard/super/system-health	SystemHealthPage	57
/dashboard/super/regime-accuracy	RegimeAccuracyPage	58
/dashboard/super/market-state	MarketStateAccuracyPage	58
/dashboard/super/exhaustion	ExhaustionTrackingPage	58
/dashboard/super/comp-entry	CompEntryQualityPage	58
/dashboard/super/trading-hours	TradingHoursPage	59
/dashboard/super/daily-profit	DailyProfitPage	59
/dashboard/super/positions	PositionManagementPage	59
/dashboard/super/smart-sl	SmartSLQualityPage	59
/dashboard/super/permissions	PermissionsPage	62

APPENDIX E / BACKEND INTERFACES

API module map

Counts of route decorators per source file, not publicly exposed or active endpoints in a deployment.

Module	Route decorators	Methods
ai_assistant	2	GET, POST
alerts	7	DELETE, GET, POST, PUT
analysis	10	GET
analytics	3	DELETE, GET, POST
auth_proxy	12	DELETE, GET, PATCH, POST
backtest	5	GET, POST
backtest_runner	9	GET, POST, WEBSOCKET
candles	5	GET, POST
comp_entry_quality	8	GET, POST
daily_profit	3	GET, POST
dashboard	5	GET
diagnostics	3	GET
ea_intel	17	GET, POST
ea_management	24	GET, POST, PUT
exhaustion	12	GET, POST
gold	6	GET, POST
levels	2	GET, POST
llm	28	DELETE, GET, POST
llm_monitor	2	GET, POST
luck	2	GET
market	5	GET, POST
market_session	1	GET
market_state_accuracy	3	GET
metatrader	27	GET, POST
ml	70	GET, POST

APPENDIX E / BACKEND INTERFACES

API module map

Counts of route decorators per source file, not publicly exposed or active endpoints in a deployment.

Module	Route decorators	Methods
news	11	GET, POST
optimizer	11	GET, POST
performance	15	GET, POST
pg_client	5	GET, POST
position_management	8	GET, POST
prompts	6	DELETE, GET, POST, PUT
push	5	GET, POST
regime_accuracy	2	GET
risk_management	16	DELETE, GET, POST, PUT
scoring_config	4	GET, POST, PUT
settings	8	GET, POST
shadow	6	GET, POST
signal_classifier	6	GET, POST
signal_engine	5	GET, POST, PUT
smart_sl_quality	5	GET, POST
sms_admin	4	GET, POST
strategies	10	GET, POST, PUT
strategy_builder	12	DELETE, GET, POST, PUT
strategy_management	12	GET, POST, PUT
system_health	2	GET
telegram	27	DELETE, GET, POST
trading	11	DELETE, GET, POST
trading_hours	2	GET, POST
websocket	1	WEBSOCKET

APPENDIX F / TRACEABLE EVIDENCE

Traceable source references

References point to a pinned commit and require repository access. E-identifiers in each chapter link to this list.

E001 AGENTS.md
E003 ArgAIAgent/main.py
E005 ArgNotificationService/app/channels
E007 ArgNotificationService/app/engine/evaluator.py
E009 ArgRegimeClassifier/app.py
E011 ArgSharpMoveIntelligence/ui/main_window.py
E013 ArgTradeAIBackend/app/analysis/comp_entry_ml/runtime.py
E015 ArgTradeAIBackend/app/analysis/comprehensive_signal.py
E017 ArgTradeAIBackend/app/analysis/ict_engine.py
E019 ArgTradeAIBackend/app/analysis/market_state_observer.py

E002 ARG_ML_EA/ARG_ML_EA.mq5
E004 ArgLicenseServer/ArgLicenseServer/ArgServer_FXAPI.py
E006 ArgNotificationService/app/engine/dispatcher.py
E008 ArgNotificationService/app/worker.py
E010 ArgRegimeClassifier/ui/main_window.py
E012 ArgTradeAIBackend/app/analysis/comp_entry_ml/features.py
E014 ArgTradeAIBackend/app/analysis/comp_entry_ml/trainer.py
E016 ArgTradeAIBackend/app/analysis/daily_profit.py
E018 ArgTradeAIBackend/app/analysis/live_scalp_engine.py
E020 ArgTradeAIBackend/app/analysis/market_state_router.py

APPENDIX F / TRACEABLE EVIDENCE

Traceable source references

References point to a pinned commit and require repository access. E-identifiers in each chapter link to this list.

E021 ArgTradeAlBackend/app/analysis/momentum_entry_policy.py
E023 ArgTradeAlBackend/app/analysis/momentum_live.py
E025 ArgTradeAlBackend/app/analysis/signal_lifecycle_v2.py
E027 ArgTradeAlBackend/app/analysis/smart_sl_quality.py
E029 ArgTradeAlBackend/app/analysis/sr_interaction.py
E031 ArgTradeAlBackend/app/analysis/trend_calculator.py
E033 ArgTradeAlBackend/app/api/ai_assistant.py
E035 ArgTradeAlBackend/app/api/backtest_runner.py
E037 ArgTradeAlBackend/app/api/diagnostics.py
E039 ArgTradeAlBackend/app/api/ea_management.py

E022 ArgTradeAlBackend/app/analysis/momentum_execution_policy.py
E024 ArgTradeAlBackend/app/analysis/position_management.py
E026 ArgTradeAlBackend/app/analysis/smart_levels.py
E028 ArgTradeAlBackend/app/analysis/sr_calculator.py
E030 ArgTradeAlBackend/app/analysis/trading_hours.py
E032 ArgTradeAlBackend/app/analysis/v2_live.py
E034 ArgTradeAlBackend/app/api/backtest.py
E036 ArgTradeAlBackend/app/api/daily_profit.py
E038 ArgTradeAlBackend/app/api/ea_intel.py
E040 ArgTradeAlBackend/app/api/exhaustion.py

APPENDIX F / TRACEABLE EVIDENCE

Traceable source references

References point to a pinned commit and require repository access. E-identifiers in each chapter link to this list.

E041 ArgTradeAIBackend/app/api/news.py
E043 ArgTradeAIBackend/app/api/risk_management.py
E045 ArgTradeAIBackend/app/api/system_health.py
E047 ArgTradeAIBackend/app/core/market_hub.py
E049 ArgTradeAIBackend/app/llm/provider.py
E051 ArgTradeAIBackend/app/ml/regime_classifier/classifier.py
E053 ArgTradeAIBackend/app/models/signal_dispatch.py
E055 ArgTradeAIBackend/app/services/signal_fanout.py
E057 ArgTradeAIBackend/app/signal/actionability.py
E059 ArgTradeAIBackend/app/signal/llm_validator.py

E042 ArgTradeAIBackend/app/api/position_management.py
E044 ArgTradeAIBackend/app/api/strategy_builder.py
E046 ArgTradeAIBackend/app/backtest
E048 ArgTradeAIBackend/app/database.py
E050 ArgTradeAIBackend/app/main.py
E052 ArgTradeAIBackend/app/ml/regime_classifier/regime_types.py
E054 ArgTradeAIBackend/app/services/ohlcv_utc_contract.py
E056 ArgTradeAIBackend/app/services/signal_feed_guard.py
E058 ArgTradeAIBackend/app/signal/dispatcher.py
E060 ArgTradeAIBackend/app/signal/unified_layer.py

APPENDIX F / TRACEABLE EVIDENCE

Traceable source references

References point to a pinned commit and require repository access. E-identifiers in each chapter link to this list.

E061 ArgTradeAIBackend/app/strategy/engine.py
E063 ArgTradeAIBackend/app/strategy/risk_policy.py
E065 ArgTradeAIBackend/app/strategy_builder/evaluator.py
E067 ArgTradeAIBackend/app/strategy_builder/state_store.py
E069 ArgTradeAIBackend/app/ws_dispatch/ws_dispatch.py
E071 ArgTradeAIBackend/mql5/ARG_ML_EA/ARG_ML_EA.mq5
E073 ArgTradeAINewsService/fetcher.py
E075 ArgTradeAITelegram/runner.py
E077 ArgTradeWebApp/src/App.jsx
E079 ArgTradeWebApp/src/contexts/AccessPolicyContext.jsx

E062 ArgTradeAIBackend/app/strategy/registry.py
E064 ArgTradeAIBackend/app/strategy/strategies
E066 ArgTradeAIBackend/app/strategy_builder/integration.py
E068 ArgTradeAIBackend/app/strategy_builder/variable_registry.py
E070 ArgTradeAIBackend/docs/production_release_contract.md
E072 ArgTradeAIBackend/mql5/ArgTradeAI_EA.mq5
E074 ArgTradeAINewsService/runner.py
E076 ArgTradeWebApp/package.json
E078 ArgTradeWebApp/src/components/strategy-builder/VisualConditionEditor.jsx
E080 ArgTradeWebApp/src/contexts/AuthContext.jsx

APPENDIX F / TRACEABLE EVIDENCE

Traceable source references

References point to a pinned commit and require repository access. E-identifiers in each chapter link to this list.

E081 ArgTradeWebApp/src/contexts/PermissionsContext.jsx
E083 ArgTradeWebApp/src/pages/dashboard/admin/RegimeAccuracyPage.jsx
E085 ArgTradeWebApp/src/pages/dashboard/admin/SystemHealthPage.jsx
E087 ArgTradeWebApp/src/pages/dashboard/super/EAMonitorPage.jsx
E089 BacktestIntelligence
E091 ParsGold_PyBridge/parsgold_client.py
E093 ParsGold_PyBridge/pg_terminals.py
E095 ParsGold_PyBridge/pg_updater.py
E097 docs/system-study-2026-09-09.md

E082 ArgTradeWebApp/src/hooks/useMarketData.js
E084 ArgTradeWebApp/src/pages/dashboard/admin/StrategiesPage.jsx
E086 ArgTradeWebApp/src/pages/dashboard/admin/TradingHoursPage.jsx
E088 ArgTradeWebApp/src/services/marketApi.js
E090 ParsGold_EA_Source/ParsGold_AutoTrade.mq5
E092 ParsGold_PyBridge/pg_manager.py
E094 ParsGold_PyBridge/pg_ui.py
E096 README.md

One product, from analysis to operation

For product evaluation and deployment scoping

ArgSrv.com

trade.argsrv.com

SOURCE REVIEW / 04 OCT 2026

production-unified

b90f26aea49d1fcb68b2eae6b509b67c61cb92a3

PRODUCT BOOK / EN / 2.0

Edition scope

This book introduces the product and mechanisms in the pinned source. Source versions do not establish installed versions across accounts.

Images and diagrams

Software images come from the actual interface running locally with synthetic data. Cover art is conceptual; diagrams explain technical relationships.

Performance evidence

No new trade, model training or profitability test was performed for this review. This catalog is not a return commitment or an audited investment-performance report.