



VEKTOR

BY
INVESTPUPPY

VEKTOR RESEARCH SERIES · 2026

Production Infrastructure: The Vektor Technical Architecture

*AWS services, architectural decisions, and design principles —
documented as a record of choices made, not a list of technologies used.*

For technical due diligence teams, CTOs, and institutional partners

IP-WP-ARCH-260501-1.0

June 2026

ABSTRACT

Most vendor technical documentation is a list of services used. This paper is a record of decisions made — including the reasoning behind each one and what it would take to change them. This paper documents the Vektor production infrastructure through that lens: the architectural decisions made at the design stage, the reasoning behind each decision, and the properties those decisions produce — modularity, scalability, and the ability to integrate with existing institutional infrastructure. The platform is currently deployed in Singapore (ap-southeast-1) for the initial SGX implementation. The architecture is region-agnostic by design: any AWS region can be configured as the deployment target without architectural change. Where capabilities are on the roadmap rather than live today, this paper says so directly.

investpuppy.com · contact@investpuppy.com · IP-WP-ARCH-260501-1.0 · Copyright 2026 InvestPuppy

1

PLATFORM OVERVIEW

What Vektor is built on, and why the infrastructure choices reflect the same design principles as the platform itself.

Vektor is deployed on Amazon Web Services using a microservices architecture. The platform is currently running in ap-southeast-1 (Singapore), chosen as the deployment base for the initial SGX implementation and for alignment with the MAS regulatory context of the founding client base.

Region-agnostic by design. Singapore (ap-southeast-1) is the current deployment region — a configuration choice, not a structural dependency. A client with UK data residency requirements deploys to eu-west-2; Hong Kong to ap-east-1; US to us-east-1 or us-west-2. AWS data residency guarantees follow the region. The two-account CI/CD architecture means deploying to a new region is a pipeline configuration, not an architecture change.

Three design principles were established before the first line of code was written. Modularity: every capability is an independently deployable service. Scalability by default: the infrastructure scales horizontally without architectural changes. Governance before growth: authentication, encryption, deployment pipeline separation, and operational visibility were designed in from the start.

Current deployment status. The architecture described in this document is designed for institutional production. The current deployment is in its pre-track-record phase, running in ap-southeast-1 with IBKR connected via paper trading. Section 9 describes which specific capabilities are live today and which are on the near-term roadmap.

2

ARCHITECTURE DIAGRAM

The complete Vektor AWS stack — nine layers, organised by function.

3

SERVICE INVENTORY

Every AWS service in the Vektor stack with its function and the decision that placed it there.

4

DATA FLOWS

How data moves through the platform from market feeds to funded positions.

FLOW 1 — DAILY MARKET DATA PIPELINE

EventBridge → Step Functions → ECS data service (equity prices: OHLCV for all configured instruments, 3-year daily history) → ECS data service (FX rates: all configured currency pairs at 8AM SGT) → RDS storage → SNS → Lambda → Slack notification on completion or failure.

FLOW 2 — RESEARCH TO FUNDED POSITIONS

Research phase: SageMaker JupyterLab — universe screening (500+ instruments → target portfolio), efficient frontier optimisation (10,000 portfolio configurations, MPT max-Sharpe selection), technical indicator selection (grid search across six indicators × 3-year history), XGBoost validation (confidence score per signal).

Strategy record: ECS writes approved strategy to RDS — instruments, weights, indicators, Sharpe ratios — pending Portfolio Manager approval.

Strategy approval: PM reviews via Admin Portal → approval record written (PM identity, timestamp, strategy ID). Maker/checker approval workflow with RBAC enforcement planned for a subsequent release; current implementation allows PM to assign directly.

Client onboarding: PM creates client profile, assigns approved strategy, sets mandate capital.

Allocation: Allocation service fetches live prices, applies lot rounding (99.94% efficiency), calculates position set. PM confirms → audit record written.

Cash funding: Operations records wire transfer (currency, amount, reference). Cash account activated.

Order execution: ECS → SQS queue → EC2 → IBKR Gateway → IBKR API (currently simulation; live before first Proof Partner client onboards). SNS notification on completion.

5

DECISION LOG

Key architectural decisions with the alternative considered and the reason for the choice made.

6

SECURITY & DATA RESIDENCY

Encryption, authentication, network isolation, and data residency guarantees.

Encryption at rest: AWS KMS for RDS and S3. Encryption in transit: all communications over HTTPS/TLS.

Authentication: Cognito with Azure EntraID SSO. Infrastructure-level RBAC (AWS IAM) planned for a subsequent release. Current enforcement is at the application layer.

Data residency: Current deployment is ap-southeast-1. AWS data residency guarantees apply per region — data does not leave the configured region under normal operation. Clients with UK data residency requirements deploy to eu-west-2. Hong Kong to ap-east-1. Switzerland (nFADP/DSG) to eu-central-1 (Frankfurt). Singapore (MAS TRM Guidelines) to ap-southeast-1. US to us-east-1 or us-west-2. Each region satisfies the data residency obligations of the applicable jurisdiction.

Network isolation: VPC with private subnets for compute and data layers. Bastion host for SSH access. API Gateway as the only public entry point.

7

SCALABILITY & REGIONAL DEPLOYMENT

How the architecture scales across users, mandates, and geographies.

Horizontal scaling: ECS services on Fargate scale independently. API Gateway handles traffic distribution. RDS scales vertically and, when required, via read replicas. No architectural changes are required to scale from 5 to 50 client mandates.

Regional deployment: the two-account CI/CD architecture makes adding a deployment region a pipeline configuration task. CodePipeline in the shared account builds Docker images, pushes to ECR, and deploys cross-account. For production client deployments, separate workload accounts per client region are the target architecture.

8

INTEGRATION ARCHITECTURE

How Vektor connects with existing DPM infrastructure.

REST API output: Vektor's allocation service exposes a REST API endpoint that returns a complete allocation table — instrument identifier, target quantity, target value, portfolio weight, and assigned indicator. A standard integration between this endpoint and an existing PMS typically takes two to four weeks for a developer familiar with the target system.

Market data substitution: Yahoo Finance is the current default data source. Bloomberg or any institutional market data feed can be substituted as the data source without architectural changes. This is a Proof Partners configuration option.

Execution venue: IBKR is the current execution venue via the IBKR Gateway and TWS API. Alternative execution venues using FIX protocol are a roadmap item. The SQS queue architecture means the execution layer is already decoupled from the rest of the application.

9

WHAT IS NOT YET IN SCOPE

Honest current state — capabilities designed but not yet live, each with a defined trigger condition.

10

CONCLUSION

Architecture as a statement of values.

The choices documented in this paper are not a list of technologies adopted because they were available. They are a record of decisions made against explicit criteria: modularity so that any component can be replaced without disrupting the others; scalability without architectural rework; governance designed in from the start rather than added in response to requirements.

The platform is in its pre-track-record phase. The architecture is designed for production. The gap between the current state and full production deployment is implementation and integration work — not architectural change. The Decision Log captures the reasoning behind each major choice so that future development decisions build on the same principles rather than cutting across them.

For AI governance and ML model management, see WP-08. For compliance architecture, see WP-06. For risk disclosure, see WP-09. All documents at investpuppy.com.

investpuppy.com · contact@investpuppy.com · IP-WP-ARCH-260501-1.0 · Copyright 2026 InvestPuppy