

■ 100LEVERAGE — LIGHT PAPER v1.0 (FINAL FULL VERSION)

AI-Powered Trading Infrastructure for Retail & Institutional Markets

0. Document Structure

Executive Summary

Vision & Mission

Core Product Pillars

Technical Architecture

Market Intelligence Layer

Razor Engine (Core AI System)

Automated Strategies

Institutional Execution Suite

Algorithmic Market

Business Model & Monetization

Security, Reliability & Compliance

KPIs & Operational Targets

Roadmap 2026

Go-To-Market Strategy

Appendix (tech stack, infra, testing, definitions)

1. Executive Summary

100Leverage is an AI-first trading infrastructure designed to deliver institutional-grade execution, real-time market intelligence, and automated trading strategies to both retail and professional users. At the core of the platform lies the Razor Engine — a multi-layer AI system that processes market data, generates validated trading signals, and executes trades through smart routing and predictive timing.

100Leverage aims to bridge the gap between manual trading tools and professional algorithmic infrastructure, offering a unified system where AI handles complexity while users retain full control over risk and strategy.

2. Vision & Mission

Vision

To create a universal AI execution layer that transforms real-time market data into actionable, validated, and institution-ready trading decisions — accessible to every trader and institution

worldwide.

Mission

Democratize algorithmic and AI-powered trading.

Eliminate execution inefficiencies for retail traders.

Provide institutions with a modular, low-latency AI infrastructure.

Build an open ecosystem where developers and quant engineers can publish AI models and strategies.

3. Core Product Pillars

3.1 Razor Engine (AI Execution Core)

The brain of 100Leverage: a multi-model execution and signal-processing engine.

3.2 Market Intelligence Layer

AI-driven real-time scanning, pattern detection, volatility forecasting, and opportunity scoring.

3.3 AI Auto-Strategies

Hands-free trading powered by Razor Engine; customizable risk profiles, strategy presets.

3.4 Institutional Execution Suite

Low-latency execution layer, enterprise APIs, OMS integration, FIX gateway.

3.5 Algorithmic Market

Open marketplace for AI models, strategy modules, developer monetization, and ranking.

4. Technical Architecture

4.1 Data Flow Overview

Data ingestion (tick, L2, OHLC, news, events)

Stream processing (parallel pipelines, feature engineering)

Model inference (ensemble of ML models)

Noise filtering (precision-focused validation)

Signal validation (fast backtest, cross-asset checks)

Execution planner (timing, slippage prediction, routing logic)

Execution layer (exchange adapters, failover, retry logic)

Feedback loop (real-time labeling, drift detection)

4.2 Latency Targets

Data ingest → signal: < 100ms

Signal → order execution: < 300ms (retail)

Institutional execution: < 50ms (p95)

4.3 Storage & Infrastructure

Tick storage: 90 days hot, 7 years cold

Feature store: ClickHouse / Redis

Model serving: TorchServe / Triton

Observability: Prometheus, Grafana, ELK

Data lake: S3 + Parquet for training

5. Market Intelligence Layer

The Market Intelligence engine powers both Razor Engine and the user interface.

Deliverables

Real-time scanning (1s resolution)

Volatility detection

Trend classification

Market event scoring

Composite risk index (0–100)

Cross-market correlation maps

Orderflow imbalance metrics

Heatmaps of liquidity depth

Performance Metrics

Event-to-alert latency: < 1s

False-alert rate: < 5%

Signal accuracy uplift vs baseline: +12–20%

6. Razor Engine (Core AI System)

6.1 Data Ingestion

WebSocket & REST connectors

Multi-venue redundancy

Latency-optimized batching

6.2 Feature Engineering

Short-term windowed features (50ms–1h)

Microstructure metrics (bid-ask spread, imbalance)

Volatility features (range expansion, ATR clusters)

6.3 Signal Model Suite

LSTM/Transformer micro-predictive models

Pattern-discovery ML models (GBDT)

Statistical alpha signals

6.4 Noise Filtering

ML-based classifier to remove false positives

Precision-optimized thresholds

6.5 Signal Validation

Multi-stage validation

Real-time backtest sim (<1s latency)

Market impact scoring

Cross-asset correlation checks

6.6 Execution Layer

Smart order routing

IOC / TWAP / liquidity-seeking algorithms

Venue selection

Slippage prediction

Failover logic + retry engine

6.7 Adaptive Learning

Auto-retrain triggers on drift

Performance monitoring

Continuous dataset expansion

7. Automated Strategies — AI Auto-Strategies

7.1 Strategy Templates

Momentum

Mean Reversion

Trend-Follow

Volatility Breakout

Market-Making (v2 optional)

7.2 Risk Profiles

Conservative

Balanced

Aggressive

Custom presets

7.3 Orchestration Engine

Entry/exit logic based on validated signals

Position sizing via volatility/risk modules

Dynamic stop-loss & take-profit

Portfolio-level constraints

Kill-switch for risk control

7.4 Deliverables

5 fully automated strategies

Pre-deployment simulation

Real-time PnL tracking

Trade logs & auditability

8. Institutional Execution Suite

8.1 API & Integration

REST API (orders, signals, data)

WebSocket streams (fills, market data)

FIX gateway (optional)

OMS integrations

8.2 Low-Latency Infrastructure

Optimized routing

Priority execution lanes

Colocation-ready endpoints

SLA 99.95%

8.3 Enterprise Tools

Audit logs

Permissioned access

IP whitelisting

Dedicated performance metrics dashboards

9. Algorithmic Market

9.1 Purpose

Create an ecosystem where developers publish trading models and strategies, and users can access them with full transparency.

9.2 Creator Pipeline

Strategy submission

Automated vetting

Sandbox backtesting

Paper-trading verification

Listing approval

Payout structure

Community ranking

9.3 Revenue Model

Creator keeps up to 70%

Platform takes 30%

Optional pay-per-call for enterprise use

9.4 Ranking & Validation

AI-curated ranking

Risk-adjusted performance scoring

Drawdown checks

Fraud-detection heuristics

10. Business Model & Monetization

Retail

Subscription tiers

Strategy marketplace purchases

Execution & automation tools

B2B / Institutional

Enterprise API licensing

Custom infrastructure

Priority execution packages

Dedicated support tiers

Marketplace

Commission on model sales

Revenue-sharing

Licensing fees for enterprise usage

Unit Economics Targets

LTV/CAC > 3

Gross margin > 60%

Enterprise churn < 5% annually

11. Security, Reliability & Compliance

11.1 Security

OAuth2 authentication

HMAC-signed API requests

AES-256 encryption at rest

TLS 1.3 in transit

Role-based access control

11.2 Regulatory Considerations

GDPR-compliant

Optional KYC for certain features

Trading audit logs

Market abuse & anomaly detection mechanisms

11.3 Reliability

Uptime SLA 99.95%

Multi-region failover

Daily incremental backups

RTO < 1 hour

12. KPIs & Operational Targets

Retail

Monthly Active Traders (MAT)

Conversion (free → paid)

Strategy AUM

User retention (30/90 days)

Institutional

Active enterprise clients

Monthly execution volume

API throughput & latency metrics

Marketplace

Listed strategies

Monthly GMV

Creator revenue distribution

13. Roadmap 2026 (Final Version)

Q1 2026 — Razor Engine v2

Enhanced execution layers

Advanced noise filtering

Multi-level signal validation

Expanded market coverage

Model ensemble refinement

Q2 2026 — AI Auto-Strategies

Fully automated strategy execution

Risk profile presets

Smart position sizing

Auto-rebalancing

Strategy performance analytics

Q3 2026 — Institutional Execution Suite

Enterprise API release

Smart routing optimizer

Latency filtering engine

OMS integration

Low-latency data pipeline

Q4 2026 — Algorithmic Market

Strategy marketplace launch

Creator tools

Monetization & payouts

AI-curated ranking

Risk-scored strategy vetting system

14. Go-To-Market Strategy

Phase 1 — Private Beta

Invite-only traders & quant partners

Collect labeled performance data

Validate execution edge

Phase 2 — Public Pro Launch

Launch Auto-Strategies

Release core Razor Engine v2 features

Begin initial marketing campaign

Phase 3 — Institutional Activation

Launch enterprise APIs

Pilot 3 institutional clients

Initiate marketplace trials

15. Appendix — Tech Stack, Testing, Infra

Recommended Stack

Streaming: Kafka / Confluent

Feature Store: ClickHouse / Redis

Data Lake: S3 + Parquet

Model Serving: TorchServe / Triton

Backend: Python (FastAPI), Go for execution

Frontend: React / Next.js

Testing

Load testing for market spikes

Latency benchmarks

Strategy performance simulations

A/B testing for execution planners

Deployment

CI/CD via GitHub Actions

Infra as Code (Terraform)

Multi-zone deployment