S I S T E M A

L A C R A

Layered Adaptive Capital & Risk Architecture

U N I F I E D W H I T E P A P E R

Architecture, Mathematics & Signal Integration

Version 3.0 — March 2026

C O N F I D E N T I A L

"The surface is a story the market tells everyone.
The structure is what the market does when no one is watching.
The weather is what's coming whether the market knows it or not."

Contents

This document covers the complete LACRA system: philosophy, mathematical foundations, signal architecture, execution pipeline, and frontend specification.

1. Executive Doctrine

1.1 What LACRA Is

LACRA (Layered Adaptive Capital & Risk Architecture) is a prescriptive capital extraction system for trading. It evaluates market conditions across three signal layers and issues direct trade signals, position recommendations, and execution structures.

The system translates institutional-grade risk analysis into plain, actionable language for retail traders. No jargon. No raw numbers except money. Second person, present tense, maximum 20 words per sentence.

1.2 What LACRA Is Not

LACRA does not predict prices. It does not forecast direction. It does not use machine learning, sentiment scraping, or social signals. It reads the terrain and tells you what you are walking into. The decision is always yours.

1.3 Core Philosophy: Machiavellian Asymmetry

The entire system follows one rule: bad news is more informative than good news.

- **φ** can degrade alignment by up to 0.30 but improve it by only 0.10
- **σ** can degrade alignment by up to 0.20 but improve it by only 0.10
- Layer disagreement can reduce clarity by up to 50% but agreement adds nothing
- The counter-view always leads with the worst signal across all layers

The cost of missing danger exceeds the cost of missing opportunity. The system trusts danger signals more than comfort signals. It treats silence from a layer as neutral, not positive.

1.4 Architectural Principles

Absence equals neutral. Missing or unavailable signals default to neutral rather than triggering alerts. This is an intentional design decision, not technical debt.

PolicyGate must never be bypassed. Every position open flows through the proposal lifecycle and PolicyGate safety layer. The gate checks friction levels, capital exposure, and regime compatibility before any capital is deployed.

Event sourcing integrity. Optimistic concurrency control, idempotent retries via UUID5, and revert patterns on failure are essential to system correctness.

Proposals revert to pending on position-open failure rather than silently succeeding.

Plain language mandate. All user-facing output is written for non-technical users. The system describes, it does not judge. Green means aligned, not good. Orange means the market works against this trade, not bad.

2. System Architecture

2.1 Pipeline Overview

The orchestrator executes a 13-step pipeline for every evaluation:

1. VALIDATE input (DecisionIntentRequest)
2. LOAD market data (OHLC via MarketDataRuntime)
3. COMPUTE ψ(t) — RegimeWaveModel + ContextEngine
4. COMPUTE φ(t) — CryptoExtractor (5 engines, 4 providers)
5. COMPUTE σ(t) — ScoutXEngine (3 classifiers)
6. COMPUTE λ(t) — layer agreement across active layers
7. COMPUTE Ω alignment — base compatibility + φ modifier + σ modifier
8. COMPUTE Ω friction — base friction + φ friction + σ friction
9. COMPUTE Ω clarity — ψ clarity × (0.5 + 0.5 × λ)
10. TRANSLATE all signals to human sentences
11. SYNTHESIZE counter-view across all three layers
12. GENERATE trade structure (entry/stop/targets/sizing) if conditions met
13. ASSEMBLE TerrainResponse + EvaluateResponse

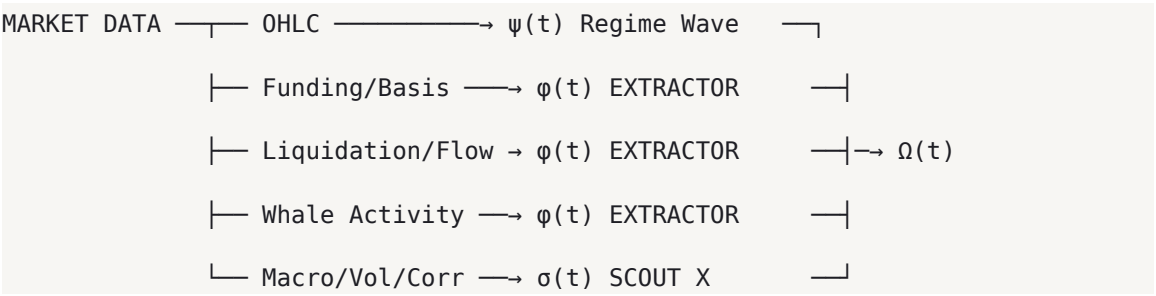
2.2 Module Map

Module	Purpose	Lines
src/regime_wave.py	Mathematical core: ψ(t) wave function, hazard, transitions	~400
src/context_engine.py	Semantic interpretation of regime state	~180
src/evaluation/friction.py	Friction engine: emotional, cognitive, execution, regret	~250
src/decision_matrix/	Compatibility scoring: intent × regime	~300
src/omega/signal_integrator.py	φ + σ modifiers, λ computation, Ω unification	~350
src/omega/trade_structure.py	Prescriptive engine: entry/stop/targets/sizing	~300
src/omega/edge_engine.py	Edge scoring	~150
src/omega/sizing_engine.py	Kelly-inspired position sizing	~200
src/omega/terrain_translator.py	Three-layer terrain labels	~200
src/crypto/extractor.py	EXTRACTOR orchestrator: 5 engines, 4 providers	~250

Module	Purpose	Lines
src/scout_x/engine.py	ScoutX: volatility, macro, correlation classifiers	~230
src/execution/service_sourced.py	Event-sourced proposal + position lifecycle	~220
src/execution/policy_gate.py	PolicyGate: safety rules for trade approval	~70
src/orchestrator.py	Main pipeline coordinator	~600

2.3 Data Flow

Three signal layers feed into a unified evaluation:



3. The Mathematical Core: ψ(t)

3.1 State Space

The market is modeled as a point s(t) flowing through a three-dimensional state space $S \subset \mathbb{R}^3$:

```
s(t) = (V, D, N)
```

V: Volatility [0, 2] — Yang-Zhang normalized volatility

D: Direction [-1, 1] — Coherence of directional movement (autocorrelation)

N: Noise [0, 1] — Market noise level (1 - efficiency)

3.2 Regime Attractors

Three regimes act as attractors in state space. Each is defined by a centroid:

Regime	V	D	N	Description
TREND	0.70	+0.50	0.25	High energy, directional, clean
RANGE	0.25	0.00	0.35	Low energy, neutral, contained
CHAOS	1.10	0.00	0.75	High energy, noisy, dangerous

3.3 Wave Function ψ(t)

The probability of each regime is computed via Gaussian kernels over the distance from each centroid:

```
ψ_β(t) = exp(-||s - μ_β||² / 2σ²) / Σ_γ exp(-||s - μ_γ||² / 2σ²)
```

Default kernel width: `σ_psi = 0.35`. The three probabilities (ψ_trend, ψ_range, ψ_chaos) always sum to 1.

Key derived metrics:

- **Entropy:** -Σ ψ_i × log(ψ_i), normalized — measures structural uncertainty
- **Dominance:** max(ψ) — how strongly one regime controls

3.4 Hazard Rate

The probability of regime change at time t:

```
h(t) = h₀ + α₁ × σ(speed/scale) + α₂ × (1 - dominance)
```

With $h_0 = 0.05$ (base), $\alpha_1 = 0.30$ (speed weight), $\alpha_2 = 0.20$ (uncertainty weight). High speed and low dominance increase the hazard. The expected holding time for the current regime is 1/h(t).

3.5 Conditional Transition

When a regime change occurs, the probability of transitioning to regime j given the current regime i is:

```
P(j | change, i) ∝ T(i→j) × A(s, j) × D(s, j)
```

T: Base transition matrix (empirical priors). **A:** Accessibility (proximity to target centroid). **D:** Drift (velocity toward target).

From \ To	TREND	RANGE	CHAOS
TREND	0.0	0.6	0.4
RANGE	0.5	0.0	0.5
CHAOS	0.4	0.6	0.0

4. The Omega Unification: Ω(t)

4.1 Three Eyes, One Terrain

LACRA v1 evaluated intent against price action alone. Omega evaluates intent against the complete terrain:

```
Ω(t) = f(ψ(t), φ(t), σ(t))
```

- **ψ(t) — The Surface:** What the price is doing (regime, entropy, hazard, stability)
- **φ(t) — The Structure:** What the plumbing reveals (funding, basis, liquidations, flows, whales)
- **σ(t) — The Weather:** What the world is doing (volatility regime, macro sentiment, correlation)

4.2 Modified Alignment

```
ψ_compat = BASE_MATRIX[action][regime]

φ_mod = compute_φ_alignment_modifier(φ(t), intent)    // [-0.30, +0.10]

σ_mod = compute_σ_alignment_modifier(σ(t), intent)    // [-0.20, +0.10]

Ω_compat = clamp(ψ_compat + φ_mod + σ_mod, 0, 1)
```

The modifiers are asymmetric by design. Good plumbing adds a little. Bad plumbing subtracts a lot. When something looks too good, the information you are missing is probably bad.

4.3 Modified Structural Clarity

```
ψ_clarity = dominance × stability × (1 - entropy)

λ = layer_agreement(ψ, φ, σ)

Ω_clarity = ψ_clarity × (0.5 + 0.5 × λ)
```

When all layers agree ($\lambda = 1$), clarity is unchanged. When layers disagree ($\lambda \to 0$), clarity drops by up to 50%. If the surface says one thing and the plumbing says another, you do not have clarity. You have an illusion.

4.4 Modified Friction

Each friction dimension now receives additive contributions from φ and σ:

```
emotional_load  = ψ_emotional  + φ_funding_stress    + σ_macro_opposition

cognitive_load  = ψ_cognitive  + φ_whale_noise       + σ_event_density

execution_risk  = ψ_execution  + φ_liquidation       + σ_liquidity_thin

regret_risk     = ψ_regret     + φ_basis_extreme     + σ_vol_mismatch
```

5. Signal Layer: φ(t) — EXTRACTOR

5.1 Design Doctrine

Every component must have a direct, traceable line to capital extraction. If it does not generate, protect, or amplify returns, it does not exist in this system.

5.2 Five Engines

Engine	Source	Output	Capital Relevance
Funding	Binance	Dislocation score, direction	Delta-neutral extraction at -6% = ~18% APY
Basis	CoinGlass	Spread %, actionable arb flag	Spot-perp arbitrage, closes by definition
Liquidation	CoinGlass	Cascade risk, asymmetry	Predicts forced selling/buying events
Flow	CryptoQuant	Net flow, direction, intensity	Exchange flow = selling pressure signal
Whale	WhaleAlert	Accumulation signal, net flow	Smart money positioning

5.3 Signal Tiers

Tier 1 — Mechanical (P0): Funding + Basis. Mechanical extraction. Near risk-free. Always available.

Tier 2 — Structural (P1): + Liquidation + Flow. Structural regime signals. Highest alpha for directional trades.

Tier 3 — Behavioral (P2): + Whale. Informational. Confirmatory signal, not primary.

5.4 Alignment Impact

Each engine computes an alignment modifier and friction modifiers:

- **Funding:** Crowd agrees with your trade and dislocation > 0.6? Alignment penalty -0.15 × dislocation. You are not with the trend. You ARE the exit liquidity.
- **Basis:** LONG in extreme contango? Optimism already priced in. Penalty -0.10 × spread.
- **Liquidation:** Cascade risk > 0.4 building in your direction? Penalty -0.20 × cascade_risk. Nuclear events get the heaviest penalty.

- **Flow:** Large exchange inflows while going LONG? Selling pressure incoming. Penalty -0.08 × net_flow.
- **Whale:** Smart money distributing while you go LONG? Penalty -0.12 × activity. Smart money accumulating while you go LONG? Bonus +0.05 × activity.

5.5 Composite Modifier Ranges

```
φ_alignment_modifier ∈ [-0.30, +0.10]   // asymmetric by design

φ_friction_modifier  ∈ [0, 0.25]        // additive per dimension
```

6. Signal Layer: σ(t) — SCOUT X

6.1 Architecture

SCOUT X provides external categorical context. It runs on every evaluation (always on). Three classifiers produce terrain descriptors:

Classifier	Input	Output Labels
Volatility	Realized volatility (V × 0.16 proxy)	low, normal, elevated, high
Macro	External sentiment	risk_on, neutral, risk_off
Correlation	Cross-asset correlation	decorrelated, normal, high

6.2 Current Status: Honest Degradation

ScoutX classifiers currently use hardcoded fallback values because the web context fetcher is not wired to live data. This produces near-constant output.

Design decision: σ is excluded from the λ (layer agreement) calculation until it carries real signal. The system reports `sigma_is_live=False` and `layers_active=["psi", "phi"]`. This is honest degradation: the system tells you exactly how many eyes are open.

To re-enable σ in the agreement calculation, set sigma_is_live=True when real data is available. One flag, one grep.

6.3 Alignment Impact (When Live)

```
σ_alignment_modifier ∈ [-0.20, +0.10]
```

- Risk-averse macro + LONG intent = -0.12
- Risk-seeking macro + LONG intent = +0.05
- Thin liquidity + any active position = execution friction +0.15
- Elevated vol + SCALP horizon = regret friction +0.15

7. Layer Agreement: λ(t)

7.1 Purpose

Layer agreement measures whether the signal layers tell the same story. It is the system's built-in bullshit detector.

```
λ(t) ∈ [0, 1]

λ = 1  →  All active layers agree

λ = 0  →  Complete contradiction
```

7.2 Computation

λ is computed as the mean of pairwise agreement scores between active layers:

- **ψ vs φ:** Does price regime agree with plumbing? Trend + extreme funding dislocation = low agreement. Chaos + extreme plumbing stress = high agreement (both say danger).
- **ψ vs σ:** Does price regime agree with macro? Trend + risk-seeking = high agreement. Trend + risk-averse = contradiction.
- **φ vs σ:** Does plumbing agree with macro? Overleveraged + risk-averse = worst case (0.15). Healthy positioning + risk-on = best case (0.85).

7.3 Active Layer Management

Only layers with real, variable signal participate in λ:

Configuration	Layers Active	λ Behavior
Crypto enabled, σ not live	["psi", "phi"]	2-layer agreement (ψ vs φ only)
Crypto disabled, σ not live	["psi"]	λ = 0.5 (neutral, insufficient data)
Both enabled, σ live	["psi", "phi", "sigma"]	Full 3-layer agreement

7.4 Downstream Impact

λ is a first-class input to:

- **Structural clarity:** Ω_clarity = ψ_clarity × (0.5 + 0.5 × λ). Low λ = clarity drops up to 50%.
- **Trade structure generation:** Trade structures are only generated when Ω_clarity meets the threshold. Low λ can prevent structure generation entirely.

- **PolicyGate:** cognitive.clarity feeds from λ. Low λ = stricter gate decisions.
- **Layer Disagreement Banner:** When $\lambda < 0.4$, the frontend shows a prominent warning.

8. Decision Matrix & Friction Engine

8.1 Base Compatibility

The decision matrix scores compatibility between the user's action and the current regime:

Action	TREND	RANGE	CHAOS
LONG	0.85	0.40	0.15
SHORT	0.85	0.40	0.15
HOLD	0.70	0.80	0.30
ADD	0.75	0.30	0.10
EXIT	0.60	0.70	0.80

8.2 Hazard Modifiers

Compatibility is further modified by hazard level and time horizon:

Horizon	Low Hazard	Medium	High
SCALP	1.0	0.9	0.7
INTRADAY	1.0	0.85	0.6
SWING	1.0	0.7	0.4
MULTIDAY	1.0	0.6	0.3

8.3 Alignment Classification

Score Range	Label	Symbol	Color
≥ 0.65	Alineada	✓	#4A9D7C (soft green)
0.40 – 0.65	Neutral	~	#D4A843 (warm amber)
< 0.40	Chocando	⚠️	#C2663B (burnt orange)

8.4 Friction Engine

Four friction dimensions, each weighted:

Dimension	Weight	Description
Emotional load	0.30	Expected emotional stress of holding the position
Cognitive load	0.25	Complexity of the decision and information environment

Dimension	Weight	Description
Execution risk	0.25	Risk of error in timing, fills, and slippage
Regret risk	0.20	Probability of regret regardless of outcome

8.5 Market Mood Classification

Condition	Label
stability ≥ 0.65 AND hazard < 0.25	calmado
stability < 0.45 OR hazard ≥ 0.40	nervioso
ψ_chaos ≥ 0.40	caótico
entropy ≥ 0.60 AND dominance < 0.50	indeciso
else	normal

9. Execution Pipeline

9.1 Proposal Lifecycle

Every position open flows through the proposal lifecycle. No shortcuts, no bypasses.

```
Evaluate → Propose → PolicyGate → Approve/Reject → Position Open
```

State transitions:

```
PENDING → APPROVED → EXECUTED → CLOSED

PENDING → REJECTED

PENDING → EXPIRED (TTL-based)

APPROVED → PENDING (on position-open failure — approval reverted)
```

9.2 PolicyGate

The PolicyGate is the system's safety layer. It evaluates proposals against configurable rules:

- Friction level checks (high/extreme friction can block)
- Capital exposure limits
- Regime compatibility requirements
- System mode enforcement (analyze / supervised / semi-autonomous / autonomous)

The gate produces one of three decisions:

Decision	Effect	System Mode
block	Proposal not created. Reason returned.	analyze
propose	Proposal created as PENDING. Human must approve.	supervised
execute	Proposal created and auto-executed. Position opened.	semi-autonomous+

9.3 Idempotency

Proposal IDs are deterministic via UUID5:

```
proposal_id = uuid5(PROPOSAL_NAMESPACE, idempotency_key)
```

Position IDs follow the same pattern:

```
position_id = uuid5(POSITION_NAMESPACE, idempotency_key)
```

This guarantees that retries with the same Idempotency-Key header return the original result instead of creating duplicates. Network timeouts and double-clicks are safe.

9.4 Failure Recovery

If a position fails to open after approval (e.g., timestamp parsing error, database conflict), the system:

14. Emits a proposal.approval_reverted event
15. Sets proposal status back to PENDING
16. Returns a 502 error with the reason
17. The frontend stays in PENDING state so the user can retry

No silent successes. No orphaned approvals.

9.5 Position Lifecycle

```
position.opened → position.closed
```

Positions track: asset, direction, size, entry/exit price, entry/exit timestamp, exit reason (TARGET / STOP / MANUAL / TIME), and outcome (PnL USD, PnL %, R-multiple, outcome class).

10. The Terrain Report

10.1 What the User Sees

The terrain report is the centerpiece of every evaluation. Three layers, all in plain language:

Surface (ψ): "The market has a clear direction with strong momentum. Conditions have been steady."

Structure (φ): "The crowd is heavily positioned on one side. Futures are pricing in significant optimism. Large holders are quietly distributing."

Environment (σ): "Volatility is compressed — a move is building. The broader market is in risk-seeking mode."

10.2 Layer Agreement in the UI

High ($\lambda > 0.7$): No special treatment. The terrain is consistent.

Medium ($0.4 < \lambda \leq 0.7$): Yellow info banner: "The price action and the underlying structure aren’t fully aligned."

Low ($\lambda \leq 0.4$): Prominent warning: "Warning: what you see on the chart is not what’s happening underneath."

10.3 The Enriched Threat Card

The counter-view synthesizes risk across all three layers:

- **Surface risk:** Regime-based risk from ψ
- **Structural risk:** Plumbing-based risk from φ
- **Environmental risk:** Macro-based risk from σ
- **Primary threat:** The single biggest risk across all layers
- **Contradiction:** Explicit when layers disagree — this becomes the headline

The threat card is always visible. It uses charcoal (#1E1E24) background with light text to visually stand apart from everything else on the page. It is never hidden, never collapsed, never optional.

10.4 Before and After

Before (isolated):

> *"Your trade matches the market. The market has a clear direction."*
>
> *Counter-view: "Trends can reverse without warning."*

After (Omega):

> *"Your trade matches the price action, but the structure underneath is strained."*
>
> *Surface: The market has a clear direction with strong momentum.*
>
> *Structure: The crowd is dangerously overleveraged. Futures are pricing in extreme optimism. Large holders are quietly distributing.*
>
> *Layer agreement: low — The price looks clean but the leverage structure is fragile.*
>
> *The biggest threat is a liquidation cascade triggered by the extreme funding imbalance. When 80% of the market is on your side, you ARE the exit liquidity.*

11. Frontend Specification

11.1 Visual Language

Clinical. Calm. Authoritative. This is a doctor’s office, not a Bloomberg terminal. The interface should feel like it knows more than it is showing. White space is confidence. Clutter is amateur hour.

11.2 Color System

State	Hex	Usage	Rule
Aligned	#4A9D7C	Verdict block background	Never means "good"
Neutral	#D4A843	Verdict block background	Means caution, not inactive
Clashing	#C2663B	Verdict block background	Market works against trade
Threat card	#1E1E24	Counter-view background	Charcoal, light text
Friction extreme	#B54A4A	Friction bars only	Only place red appears
Friction low	#6B8FA3	Friction bars only	Neutral, not good
Page surface	#F7F6F3	Background	Warm off-white

11.3 Result Screen Hierarchy

18. Intent Summary (always visible)
19. Verdict Block — full-width colored block reflecting Ω alignment
20. Warnings banner (if any)
21. Layer Disagreement Banner (if $\lambda < 0.4$)
22. Counter-view / Threat Card (charcoal, always visible)
23. Trade Structure Panel (entry/stop/targets/sizing + PolicyGate flow)
24. Terrain Section (three-layer, always visible)
25. Expandable sections: Friction, Capital Impact, Thinking Frame, Interpretation
26. Disclaimer (always visible, non-dismissable)

11.4 Hard Rules

- No raw floats ever. No entropy: 0.35, no psi_trend: 0.78, no compatibility_score: 0.81.
- Counter-view and disclaimer are always visible, no exceptions.

- Green means aligned. Not good. The fit_statement clarifies.
- Red only appears for extreme friction. Never for alignment.
- No bright saturated colors anywhere. Everything muted. Confidence is quiet.

12. API Reference

12.1 Core Endpoints

Method	Path	Description
POST	/v1/evaluate	Full Omega evaluation pipeline
POST	/v1/plan	Playbook selection + execution plan
GET	/v1/playbooks	Playbook catalog (PB1-PB6)
GET	/v1/health	Health check

12.2 Execution Endpoints

Method	Path	Description
POST	/v1/execution/proposals/from-evaluation/{id}	Create proposal from evaluation (PolicyGate)
POST	/v1/execution/proposals/from-plan	Create proposal from plan
GET	/v1/execution/proposals	List proposals
POST	/v1/execution/proposals/{id}/approve	Approve + open position
POST	/v1/execution/proposals/{id}/reject	Reject proposal
GET	/v1/execution/positions	List positions
POST	/v1/execution/positions/{id}/close	Close position with outcome
GET	/v1/execution/mode	Get system mode
POST	/v1/execution/mode	Set system mode (admin)

12.3 Evaluate Response Structure

The complete EvaluateResponse includes:

- `evaluation_id`, `timestamp`, `asset`, `intent_summary`
- `terrain`: { surface{}, structure{}, environment{}, layer_agreement, layer_agreement_statement, layers_active[], layers_count }
- `decision_fit`: { alignment, alignment_symbol, fit_statement }
- `friction`: { overall_level, emotional_load, cognitive_load, execution_risk, regret_risk, friction_statement }
- `capital_impact`: { statement, expected_volatility, size_assessment }
- `thinking_frame`: { core_insight, what_goes_wrong[], what_tends_to_work[] }
- `interpretation`: { analyst_view, human_view }

- `counter_view`: { statement, surface_risk, structural_risk, environmental_risk, primary_threat, watch_for[], contradiction }
- `trade_structure`: { has_structure, entry_zone{}, stop_loss{}, target_primary{}, target_secondary{}, invalidation_conditions[], sizing{}, edge{} }
- `warnings[]`, `disclaimer`, `meta{}`

13. Operational Reference

13.1 Environment Variables

Variable	Required	Description
LACRA_MARKET_API_KEY	Yes	TwelveData API key for market data
LACRA_EXECUTION_API_KEY	Yes	Secret key for execution endpoints
CRYPTO_EXTRACTOR_ENABLED	No	Enable φ(t) crypto signal extraction
BINANCE_API_KEY	If crypto	Binance API key (funding rates)
COINGLASS_API_KEY	If crypto	CoinGlass API key (liquidations, basis)
CRYPTOQUANT_API_KEY	No	CryptoQuant (exchange flows)
WHALE_ALERT_API_KEY	No	WhaleAlert (large transactions)
LACRA_ALLOW_LEGACY_CSV	No	Enable CSV data fallback
LACRA_ALLOW_YFINANCE	No	Enable yfinance fallback
LACRA_H_ENABLED	No	Enable LACRA-H extended evaluation
LACRA_SYSTEM_MODE	No	Default system mode (supervised)

13.2 Startup

```
# Backend (from project root)

python runners/serve.py              # API on localhost:8000


# Frontend

cd frontend && npm run dev            # UI on localhost:3000
```

13.3 Playbook Catalog

ID	Name	Trigger Condition
PB1	Range Edge Reversal	Range regime, low-medium transition risk, clear structure
PB2	Failed Breakout / Trap Fade	Range regime, high transition risk
PB3	Trend Pullback Continuation	Trend regime, clear structure, not high transition risk
PB4	Breakout With Confirmation	Clear structure, low risk, calm mood, LONG/SHORT only
PB5	Transition Scout Mode	High transition risk in any regime
PB6	No-Trade Protocol	Hostile environment, chaotic mood, or extreme friction

13.4 Codebase Metrics

Category	Files	Lines
Source (src/)	110	~22,500
Tests (tests/)	43	~12,500
Frontend (frontend/)	20	~3,400
Total	~220	~40,000

13.5 Do-Not-Fix List

These are intentional design decisions, not technical debt. Future contributors must not "correct" them:

- **Friction in both DecisionMatrix and FrictionEngine:** Intentional cross-validation, not redundancy
- **absence-equals-neutral:** Missing signals default to neutral, never trigger alerts
- **Machiavellian asymmetry in modifiers:** Bad news subtracts more than good news adds
- **sigma_is_live=False:** Intentional exclusion until ScoutX has real data
- **PolicyGate is never bypassed:** Even if it feels slow or restrictive

— END OF DOCUMENT —

Sistema LACRA v3.0 — March 2026