Narayan Jyotish Seva — Signature Intelligence Platform

HIGH-LEVEL DESIGN · V1.4 · AI AUDIENCE EDITION

JUNE 07, 2026

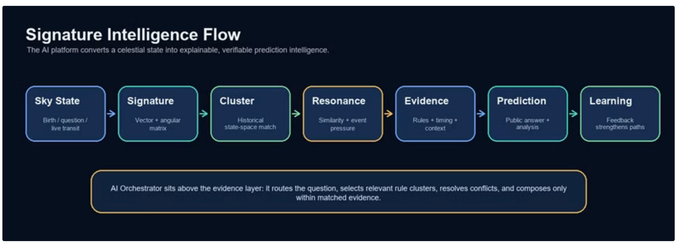
A Signature Intelligence Platform powered by 550 years of celestial state-space indexing, resonance clustering, explainable rule orchestration, and feedback-driven prediction learning.

Built on **30,000+ deterministic rules**, our rule-based AI orchestration delivers human insight grounded in facts, not jargon. By binding scope to evidence, the platform produces practical, trustworthy predictions. Based on space, time, and intent axes, the platform leverages feedback for runtime alignment with user needs—delivering predictions that evolve with context.

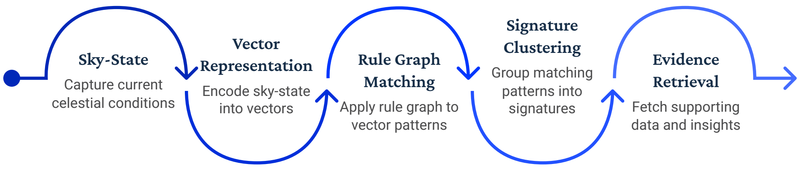
The Foundation: Rule Graph, Vector Space & Signature Architecture

The platform combines a Rule Graph, a Vector Space, and Signature Architecture to turn celestial conditions into actionable insight. The flow stays practical: transform the sky-state, match the right rules, then retrieve the most relevant evidence.

This complete 7-stage pipeline shows how the platform moves from celestial input to explainable output.



This complete 7-stage pipeline shows how the platform moves from celestial input to explainable output.



How the platform works in practice

At a practical level, the system moves from observation to interpretation in a disciplined sequence. It does not jump directly from a celestial condition to a prediction. Instead, it first normalizes the sky-state, then translates it into a machine-readable vector form, and only then applies rule logic and clustering to identify the strongest evidence trail.

1. Sky-state capture. The process begins with the raw celestial condition at a specific moment. This is the input layer: the system reads the relevant planetary positions, relationships, and timing context as structured state, not as a loose narrative. That matters because every downstream decision depends on having a precise, consistent representation of what the sky is doing now.

2. Vector representation. Next, the sky-state is encoded into a vector space. In simple terms, the platform converts symbolic astronomical conditions into a format that can be compared, scored, and searched efficiently. This step is what allows the system to move beyond static lookup and into similarity-based reasoning.

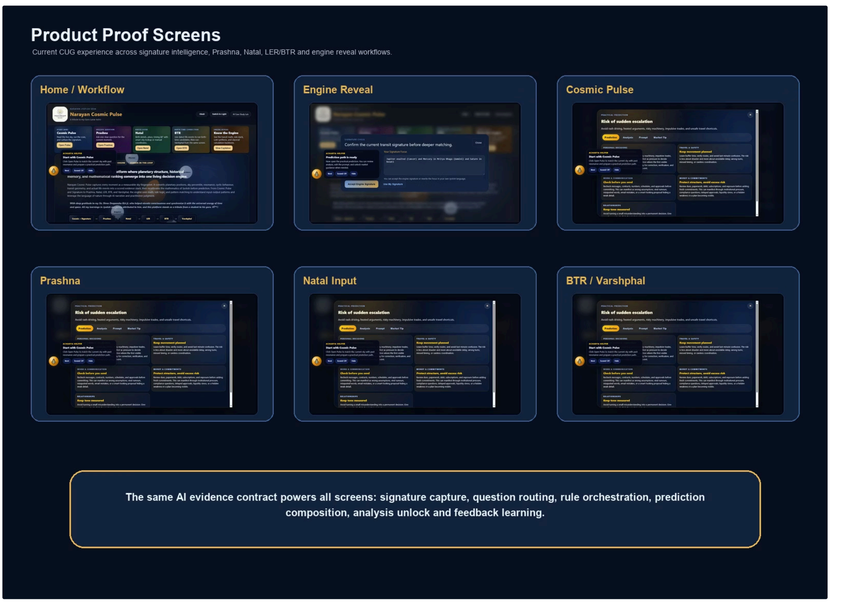
3. Rule graph matching. The vectorized state is then matched against the rule graph. The rule graph acts like a structured map of deterministic logic: it identifies which conditions, combinations, and thresholds are relevant. This is the point where the platform applies domain knowledge in a consistent and auditable way.

4. Signature clustering. Once the system finds candidate matches, it groups them into signatures. A signature is not just a single hit; it is a coherent pattern of evidence. Clustering matters because the strongest conclusion usually comes from multiple reinforcing signals, not from one isolated rule.

5. Evidence retrieval. Finally, the platform retrieves supporting evidence for the best-matching signatures. This is where abstract scoring becomes operationally useful: the system surfaces the actual support metrics behind a conclusion so the result can be reviewed, defended, and used with confidence.

Unified workflow coverage and evidence proof

The product proof screens show all six workflows operating on the same underlying rule-based orchestration engine: Home/Workflow, Engine Reveal, Cosmic Pulse, Prashna, Natal Input, and BTR/Varshaphal. Each screen uses the same evidence contract across the journey — signature capture, question routing, rule orchestration, prediction composition, analysis output, and feedback learning.



Across these workflows, the interface changes to fit the use case, but the engine does not. Every screen is driven by the same deterministic matching layer, the same signature framework, and the same evidence retrieval logic. That consistency is what makes the platform scalable: one architecture, multiple experiences, one operating standard for trust.

The evidence detail view then makes that contract visible at the lowest level. It shows the actual scoring behind each prediction, including specific angle support values such as **Sa-Ju**, **Ve-Sa**, and **Ma-Ra**, plus cluster-level metrics like **Cluster similarity**, **Sky spread**, and **Nak bonus**. Together, these scores explain why a signature was selected and how strongly the underlying evidence supports it.

In executive terms, the proof screens demonstrate product breadth, while the evidence detail view demonstrates product depth. One proves that the architecture powers multiple workflows; the other proves that each output is grounded in transparent, auditable scoring. That is the core value of the platform: every workflow is not only consistent, but explainable end to end.

Executive Design Summary

Event-Based Prediction · Timing Framework · Runtime Feedback Incorporation

Narayan Cosmic Pulse delivers event-based prediction for executive decision support. It converts structured user inputs into practical timing insights for major life events, incorporates runtime feedback as new outcomes are entered, and produces explainable guidance that can be reviewed and acted on with confidence.

The system is designed to support high-stakes planning: it estimates probable windows for key life events such as marriage, childbirth, first job, and career changes, while allowing actual events to be entered at runtime so the model can benchmark and refine future guidance. The result is a practical framework for aligning predicted windows with observed outcomes and improving decision quality over time.

01

Event Capture & Analysis

Birth chart, question, transit moment, or event reference captured as a structured input for analysis

02

Timing Framework Estimation

Probable windows estimated for major life events using the timing framework

03

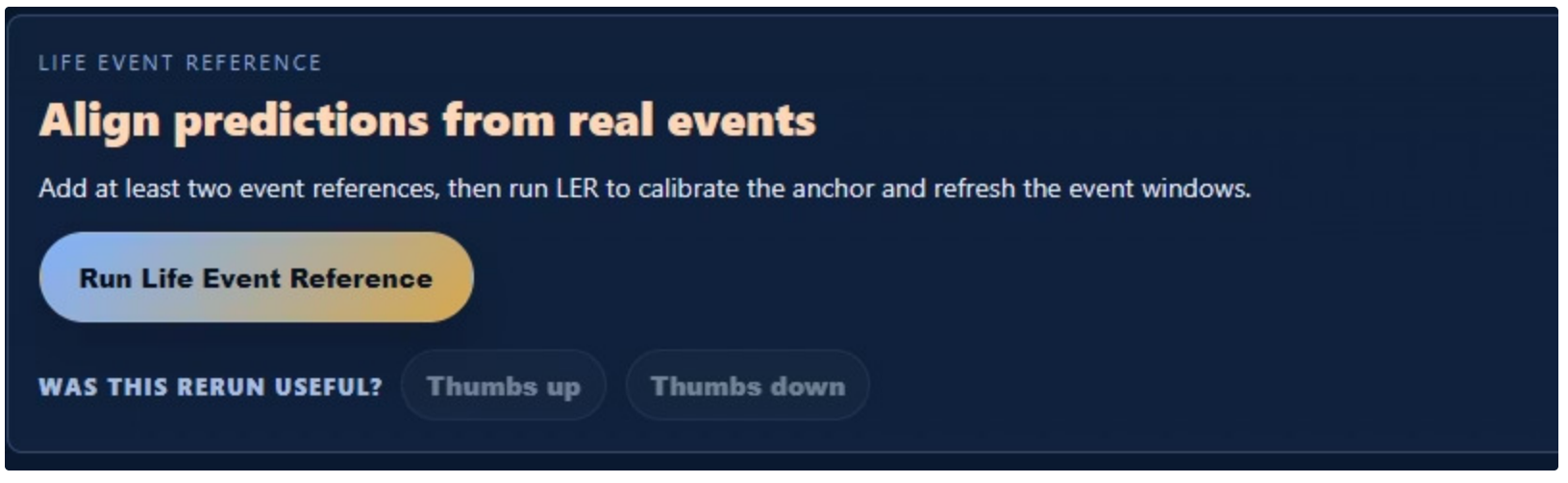
Runtime Feedback Integration

Actual user-entered outcomes incorporated to benchmark and calibrate future estimates

04

Explainable Prediction Output

Traceable evidence and timing guidance assembled into a clear, executive-ready output



This architecture directly addresses hallucination risk in specialist domains. By constraining the system to event-based prediction, timing analysis, benchmarking, and traceable feedback incorporation, it produces outputs that are reproducible, inspectable, and grounded in domain logic rather than unconstrained LLM generation. It also extends beyond prediction into market tips of the day, making the engine useful for both long-range life-event timing and daily intelligence.

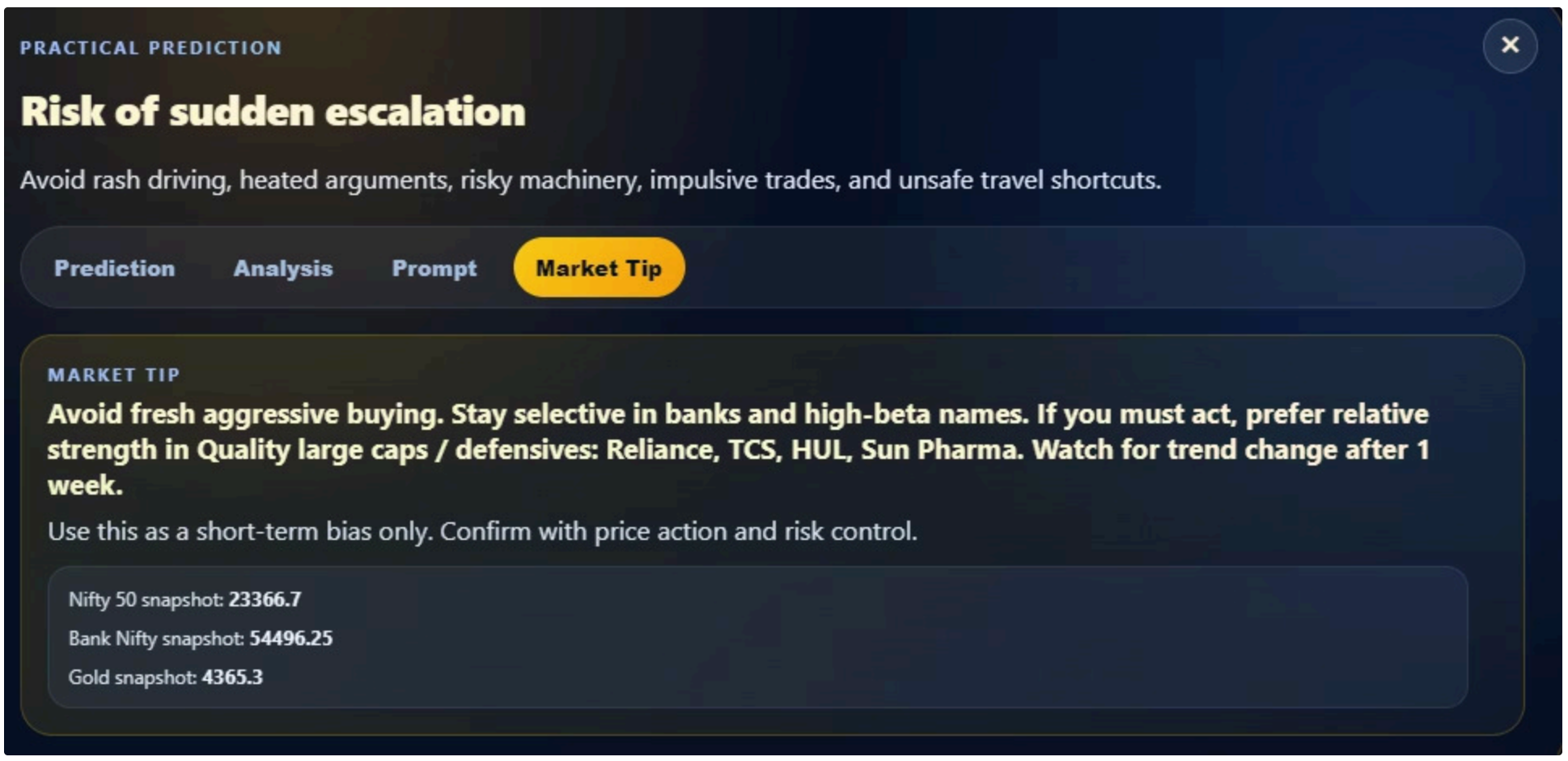
Orchestrator: Evidence Coordination & Conflict Resolution

The Orchestrator is the **reasoning coordinator**. It gathers evidence from multiple rule engines, compares conflicting signals, and assembles a final answer from the strongest supported path.

> **DOMAIN VARIABLES** are configurable inputs and outputs that let the same reasoning system operate across different domains. The orchestrator stays the same; only the variables and rule sets change.

Key Domain Variables in the Current Implementation

- **Query Topic** — what the user is asking about
- **Primary Anchor** — the main reference point
- **Secondary Anchors** — alternative reference points
- **Evidence Clusters** — grouped rule matches
- **Timing Witnesses** — multiple timing signals
- **Confidence Levels** — evidence quality ranking



Market Tip Example: How the Same Orchestrator Works

In a market tip, the **Query Topic** may be **“Risk of sudden escalation”**. The orchestrator then treats market levels and related instruments as the reference points for reasoning. **Nifty levels** become the **Primary Anchor**, while **Bank Nifty** and **Gold** act as **Secondary Anchors** that help confirm or challenge the main view.

The next step is evidence collection. Instead of relying on a single signal, the system builds **Evidence Clusters** from multiple market inputs such as price action, volatility shifts, breadth, sector strength, and cross-asset behavior. These clusters are compared against each other so the orchestrator can identify the strongest direction and filter out weak or conflicting interpretations.

Timing Witnesses then validate whether the expected move fits the current window. If the evidence suggests pressure is building, the timing signals confirm whether that setup is immediate, delayed, or not yet actionable. Finally, **Confidence Levels** determine how strongly the market tip should be expressed — for example, whether the conclusion is high-confidence, tentative, or only worth monitoring.

Why This Matters

This same orchestrator can support life events, market predictions, or any other domain. The reasoning engine does not change — only the domain variables, anchors, evidence sets, and timing rules are swapped in for the task at hand.

Four Core Product Features

Each feature is built on the same execution pattern: **capture input state, derive evidence, apply rule matching, and generate an output with traceable reasoning**. The features differ in the input signal they accept, but they share the same technical contract and feedback loop.

Cosmic Pulse

What it does: Market intelligence and daily tips. **Input:** Market state and current signals. **Output:** A concise, evidence-backed market tip.

Prashna

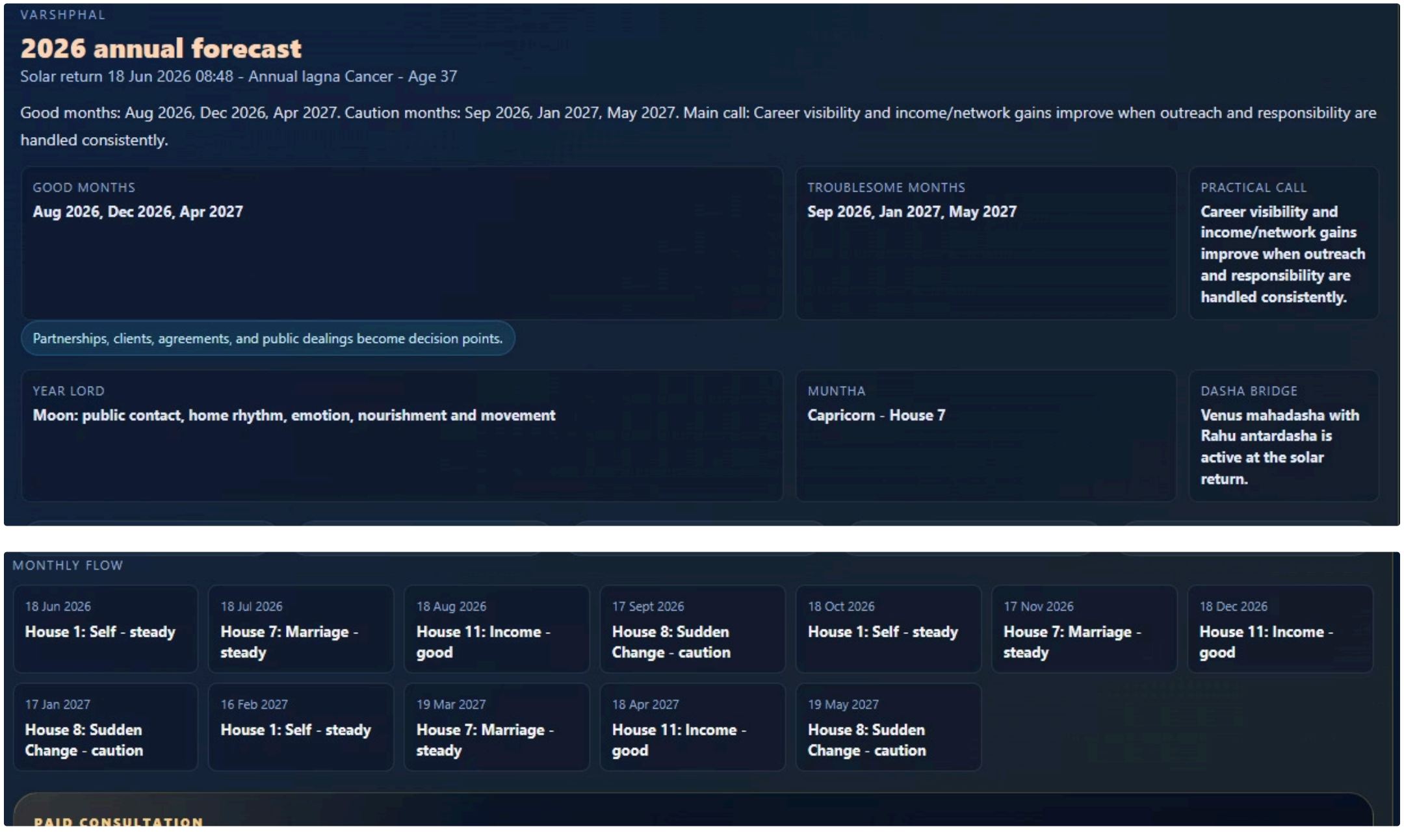
What it does: Question-based prediction using the live sky state. **Input:** User question plus current sky state. **Output:** A direct, evidence-based answer.

Natal

What it does: Birth chart analysis and life event timing. **Input:** Birth data and event markers. **Output:** A prediction with supporting reasoning and timing cues.

BTR & Varshaphal

What it does: Birth time rectification and annual forecasts. **Input:** Life events and yearly chart context. **Output:** A revised birth time estimate and annual forecast.



Learning Loop: Continuous Refinement

The learning loop is the engine's self-improvement mechanism: every prediction is compared against reality, every verified outcome becomes a training signal, and every signal is used to refine the rule graph. In effect, the platform does not simply produce answers — it compounds knowledge with each interaction, becoming more precise, more reliable, and more defensible over time.

01

User verification

The user confirms the actual outcome against the predicted outcome, creating a clear ground-truth reference for learning.

02

Training signal capture

Verified outcomes are recorded as structured feedback, turning each interaction into reusable system input.

03

Rule strengthening

Rule paths that produced correct predictions are reinforced, increasing their influence in future reasoning.

04

Rule demotion

Rule paths that underperformed are reduced in weight or removed, improving signal quality and reducing error propagation.

05

Evidence recalibration

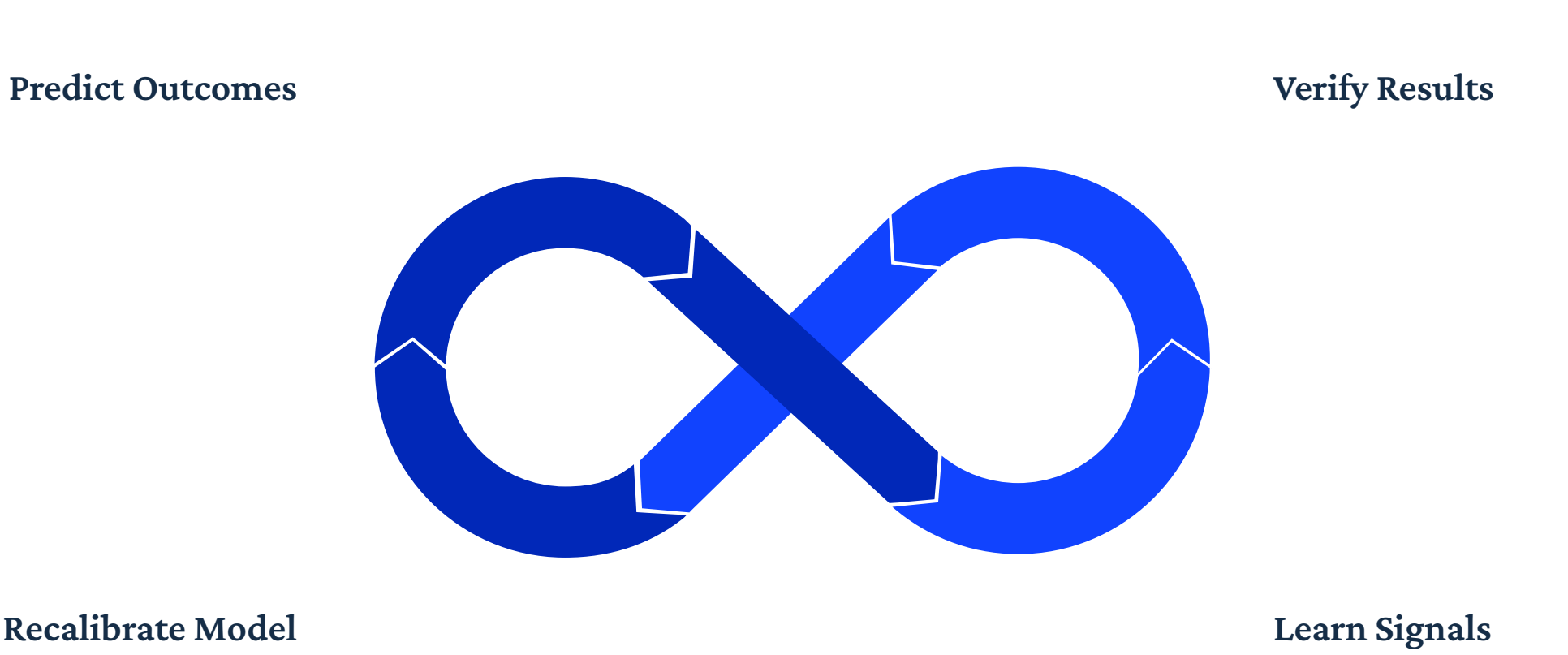
Evidence weights are continuously updated so the graph reflects the latest validated patterns and higher-confidence paths.

Why This Matters

- Creates a defensible competitive moat by compounding proprietary feedback over time.
- Each user interaction improves the system, making the product more valuable with every prediction.
- Proprietary feedback data becomes increasingly valuable as the rule graph grows richer and more accurate.
- The system gets smarter with scale, because usage continuously improves the underlying model of decision-making.
- Over time, the platform becomes measurably more accurate, more trusted, and harder to replicate.

> Every verified prediction strengthens the system. Every failed prediction teaches it what to avoid. That is the core of the learning loop.

Feedback Loop Cycle



This cycle ensures the platform improves continuously. Predictions are tested against outcomes, outcomes become training signals, and the resulting rule updates improve the next round of predictions. The result is a self-reinforcing system that becomes stronger, sharper, and more accurate with every verified interaction.

Why This Matters: The Competitive Advantage

In an era increasingly shaped by AI, the distinction between generic fluency and domain-specific precision is paramount. Our platform addresses the critical shortcomings of conventional AI in specialized domains, transforming unreliable outputs into trusted, auditable insights.

The Problem: AI Hallucination

Generic large language models falter in specialist fields, generating plausible but often unfounded information. This lack of explainability and frequent hallucination renders them unsuitable for applications demanding high accuracy and verifiable reasoning.

The Solution: Deterministic Orchestration

We deliver verifiable outcomes through rule-based orchestration and deterministic evidence composition. Our system moves beyond probabilistic guesswork, ensuring every prediction is grounded in a transparent, logical framework.

The Moat: Signature Intelligence & Learning Loop

Our proprietary "Signature Intelligence Layer" indexes complex sky-states into comparable clusters, enabling instant retrieval from a dynamic graph of over 30,000 rules. This, coupled with an inherent self-improving learning loop, creates a robust and defensible technical advantage.

The Result: Reproducible, Auditable Predictions

The platform consistently produces reproducible, auditable, and continuously refined predictions. Every output is backed by traceable evidence, eliminating the opaque "black box" and fostering unparalleled trust and reliability.

Our Positioning: AI-Native Reasoning

This is not a data visualization tool or a mere chart display application. We are building an AI-native reasoning system, engineered for profound analytical depth and unparalleled domain-specific intelligence, setting a new standard for AI applications.

Technical Architecture: System Components & Evidence Pipeline

Our platform is built upon a modular architecture, where specialized components work in concert to process information and derive insights. This design ensures flexibility, scalability, and maintainability, with each module contributing to a robust evidence pipeline that culminates in auditable predictions.

Core System Components

These are the foundational modules that capture inputs, derive evidence, orchestrate AI reasoning, and generate outputs, all while integrating with the continuous learning loop.

cosmic_pulse_api_v6.py
Current sky, historical scan and signature outputs

cosmic_resonance_engine.py
Resonance scoring and sky-pattern comparison

natal_chart_context.py
Chart context, vargas, cusps, progressions

ai_prediction_orchestrator.py
Main question AI orchestration

prediction_evidence_registry.py
Rule evidence collection before output generation

rule_corpus_index.py
Structured evidence search across 30,525 rule rows

rule_based_prediction_generator.py
Public prediction composer using the 25% framework

feedback_api.py
Feedback capture and learning loop integration

Evidence Corpus Structure

The core of our reasoning engine is powered by a vast and continuously growing corpus of rules and evidence, meticulously categorized for precise retrieval and application.

Category	Number of Rule Rows
Central Jyotish Rule Engines	13,871
Prashna Engines	8,882
Navamsha Engines	2,692
BPHS Metadata	1,393
Nadi/BNN/Progression	1,260
Nakshatra Engines	771
Other sources	1,500+
Total	**~30,525 searchable rule/evidence rows**

Evidence Cluster Strategy

Our system employs a sophisticated clustering strategy to query and synthesize evidence, ensuring a comprehensive and nuanced analysis for every prediction.

- House/Sign/Karaka clusters for base promise
- Nakshatra clusters for behaviour and micro-prediction
- Varga clusters (D9, D7, D10, D2, D4, D16) for promise confirmation
- Timing clusters (Yogini, transit, progression, dasha) for windows
- Relative House clusters for derived questions
- Vastu/Space clusters for physical clues
- Feedback clusters for learning

API Endpoints

Endpoint	Description
/cosmic-pulse	Live cosmic pulse scan
/cosmic-pulse/predictions	Cosmic prediction and historical match
/prashna/evaluate	Prashna workflow
/natal/evaluate	Natal workflow
/btr/evaluate	Birth-time rectification
/varshaphala/evaluate	Annual forecast
/payment-intents	Payment/pass creation
/engine-feedback	Feedback capture
/consultation/request	Paid consultation request

Data & Feedback Architecture

Data Layer	Storage / Implementation
User submissions	SQLite through submission_store.py
Engine outputs	JSON payload stored with submission
Feedback	SQLite engine_feedback
Fired rules	Stored in analysis/feedback payloads
Rule corpus	JSON, JSONL, CSV and Python engines
Static UI/media	backend/static, docs, media folders
Ephemeris	Local ephe folder

1. Prediction is shown
2. User marks Yes/No/Test later with optional comment
3. Public output, rule evidence and analysis payloads are saved
4. Owner can review and mark rule path verified or unverified
5. Future tuning can increase/decrease rule confidence

This modular architecture, coupled with a deep and structured evidence corpus, enables the platform to be remarkably domain-agnostic. The same core components can utilize different rule sets, allowing for versatile application while maintaining consistent precision and reliability across diverse specialized fields.

NARAYAN COSMIC PULSE

Consciousness Making Reality

चेतना से चमत्कार

About Narayan Cosmic Pulse

Narayan Cosmic Pulse was created by Nupendra Joshi, a tech enthusiast passionate about merging ancient wisdom with cutting-edge AI, grounded in a deep understanding of both the domain and the underlying technology.

I’m solving the critical challenge of bringing domain-specific precision and auditable insights to complex fields where generic AI often falls short, turning unreliable outputs into trusted, verifiable reasoning.

My vision is to help individuals and organizations with an AI-native reasoning system that delivers reproducible, auditable, and continuously refined predictions, setting a new standard for intelligent decision-making and building real trust along the way.

This platform represents years of research into bridging domain expertise with AI precision, and it matters to me personally because I believe the right intelligence should feel both deeply capable and genuinely trustworthy.

nupendra@narayancosmicpulse.com · [Your website URL] · Watch the engine in action

Contact Us | Learn More