TECHNICAL WHITEPAPER — VERSION 1.1

Presigate™: A Pre-Flight Decision Gate for Autonomous AI Agents

Condition-readiness infrastructure for the agentic AI economy

A five-primitive framework — RXI™, MMI™, CSI™, TVI™, GSI™ — grounded in peer-reviewed academic literature and validated against documented real-world failure cases, establishing the condition-gating layer that autonomous AI agent execution requires.

ORGANIZATION

YodaCom LLC
A YodaCom company

CLASSIFICATION

External — Technical Partners
NVIDIA Collaboration

DATE

Revised July 21, 2026
Orig. June 20, 2026 · Version 1.1

Abstract

Autonomous AI agents are transitioning from recommendation engines to execution systems. They hold real wallets, commit real resources, and take irreversible actions — buying, transacting, calling tools, moving money — without human approval in the loop. Yet no standard infrastructure layer exists to verify whether external conditions are right for an agent to act before it fires. This paper introduces Presigate, a pre-flight condition-gating framework: a general-purpose gate that sits synchronously inside an autonomous agent's execution path and answers one question before any consequential action fires — are conditions right to act, right now? The framework is demonstrated in depth through five named intelligence primitives built for the trading domain — RXI (Regime eXecution Intelligence), MMI (Market Microstructure Intelligence), CSI (Composite Sentiment Index), TVI (Trust Verification Index), and GSI (Gate Signal Index), the composite integrator — and, as this revision documents, the same gate architecture has since been reduced to practice for two further domains: a seven-primitive tool-safety family for AI-agent tool calls, and a settlement-rail condition primitive for stablecoin and payment-rail value transfers (Sections 3.6–3.7). Each trading primitive is grounded in peer-reviewed academic literature and validated against documented real-world failure cases where acting without that condition check produced quantified, named losses. The oracle precedent from decentralized finance — where an identical structural gap (unverified data inputs to autonomous contracts) produced hundreds of millions in losses before the industry mandated a solution — provides the adoption curve argument: condition-gating will follow the same path from optional to mandatory infrastructure. Presigate is designed to be that layer, for any autonomous action — not only crypto trading. The core architecture described in this paper is patent pending (U.S. provisional application filed July 2026).

1. The Problem: Autonomous Agents Are Executing Without Environmental Awareness

1.1 The Shift from Recommendation to Execution

The AI agent economy is no longer theoretical. As of May 2026, Amazon Web Services has issued actual cryptocurrency wallets to AI agents, enabling autonomous discovery, negotiation, and payment execution within a single loop — no human approval required [Fime, 2026]. Coinbase has launched wallet infrastructure specifically designed for agents to spend, earn, and transact autonomously at enterprise scale. The x402 open payment protocol has processed over 50 million AI-to-AI transactions.

Market scale projections from major research firms reflect the acceleration. Mordor Intelligence (2025) values the agentic AI market at $6.96 billion in 2025 and projects $57.42 billion by 2031 at a 42.14% CAGR. Grand View Research (2025) projects $182.97 billion by 2033. McKinsey (October 2025) estimates that AI agents mediating consumer purchasing could represent $3 to $5 trillion of global consumer commerce by 2030, with US B2C retail alone at $900 billion to $1 trillion [McKinsey, 2025]. These are projections, not facts — but the directional consensus across independent research firms is consistent: autonomous agent deployment is accelerating at a rate that makes safety infrastructure a near-term architectural necessity, not a long-term consideration.

Enterprise adoption is early but moving. Sixty-one percent of CEOs report integrating agents into core operations. AI agents have demonstrated the ability to reduce human task time by up to 86% in multi-step workflows, and 88% of early adopters report positive ROI [Mordor Intelligence, 2025].

1.2 The Safety Evaluation Gap

Deployment pace has outrun safety infrastructure. The 2025 AI Agent Index — a systematic study of 30 deployed agentic systems by MIT researchers (arXiv:2602.17753, February 2025) — found that of the 13 agents exhibiting frontier levels of autonomy, only 4 disclose any agentic safety evaluations. Twenty-five of 30 agents share no internal safety results; 23 of 30 have undergone no third-party testing [MIT AI Agent Index, 2025].

The documented failure cases are already on the record. In early 2025, a commercial AI agent tasked with checking egg prices instead purchased them without user consent. An AI coding assistant asked to organize a folder moved files such that neither agent nor operator could locate them afterward. Anthropic's stress-testing of 16 frontier models found that agentic misalignment — agents independently choosing harmful actions — emerges across all major providers [Adversa AI, 2025].

1.3 The Specific Gap This Paper Addresses

The existing agent safety ecosystem addresses two well-defined questions:

- **What should the agent say?** (Content safety, output filtering, topic guardrails)
- **What is the agent permitted to do?** (Access control, policy enforcement, permission frameworks)

No current standard addresses a third, distinct question:

- **Are external conditions right for the agent to do this, right now?**

This is the condition-readiness gap. NVIDIA NeMo Guardrails — the leading open-source toolkit for programmable LLM safety — implements input rails, output rails, dialog rails, retrieval rails, and execution rails [NeMo Guardrails, arXiv:2310.10501]. Its purpose is to prevent harmful content generation and enforce behavioral boundaries. It does not evaluate whether market state, data freshness, liquidity depth, or environmental regime make a given moment the right one for execution. No existing production tool does. The frameworks are built by ML engineers focused on model behavior; Presigate is built by quantitative finance practitioners focused on execution environment. The two problems are complementary but orthogonal.

This gap is not specific to financial trading. Any autonomous agent that can buy, sell, transact, call a tool, or move money without a human clearing the action first faces the identical structural problem: authorization to act is not the same question as whether conditions are right to act, right now. This paper demonstrates the condition-gating framework in the trading domain first, because trading is where the evidence is most immediate, most quantified, and most dollar-denominated — but the underlying architecture is domain-general. Sections 3.6 and 3.7 document two further domains — AI-agent tool calls and settlement-rail value transfers — where the same gate, unmodified in its underlying design, has since been built and is live in production.

The cost of this gap is already documented. The remainder of this paper presents the evidence, the framework, and the architecture of the solution.

1.4 The Scale Problem: Correlated Agent Execution and Systemic Risk

The per-agent framing of the condition-readiness gap — one agent, one bad decision, one recoverable loss — is incomplete. It describes the floor, not the ceiling, of the risk surface.

The more significant risk is correlated. Autonomous agents trained on the same data, exposed to the same signals, and wired to the same execution venues do not make independent decisions. They make the same decision at the same moment. When thousands of agents execute simultaneously under the same condition signal, they do not simply add — they amplify. The cascade they collectively trigger is larger than the sum of their individual actions.

This is not a theoretical concern. The canonical historical case is the 2010 Flash Crash. On May 6, 2010, correlated algorithmic trading programs — each individually rational, each operating within its own risk parameters — executed in concert and triggered a 1,000-point intraday decline in the Dow Jones Industrial Average. Nearly $1 trillion in US equity market value evaporated temporarily. The CFTC-SEC joint staff report documented the mechanism precisely: correlated autonomous execution into thinning liquidity, with no cross-system visibility and no pre-execution gate [CFTC-SEC Joint Report, 2010]. No individual system caused the event. The correlation did.

The agentic AI economy replicates this mechanism at greater speed, lower latency, and expanding scope. The structural conditions are identical: shared training data, shared signal exposure, shared execution surfaces. The failure mode is not new — but the acceleration and scale are.

The gap the per-agent condition check cannot close. A gate that evaluates one agent's execution conditions before it fires cannot detect whether a thousand agents are about to fire simultaneously on the same signal. Per-agent condition checks are necessary but not sufficient for systemic safety. The missing layer is a cross-agent visibility function: infrastructure that can see the shape of the agent population's collective execution intention before it crystallizes into action, and return a condition signal that individual agents can act on before the cascade amplifies.

Presigate's architecture toward this. The per-agent pre-flight gate is the current validated capability and the real product. The systemic-safety layer — cross-agent visibility enabling cascade-dampening across the agent population — is the trajectory the data flywheel enables. Each

integrated agent is a sensor. Each pre-execution query is a data point. As the cross-agent corpus accumulates, the gate's signal gains cross-agent context. The per-agent gate becomes the distribution mechanism through which systemic safety is delivered. This is the honest framing of the direction: not a shipped feature, but the logical destination of the architecture.

2. The Oracle Precedent: When Autonomous Systems Needed a Verified-Conditions Layer

The argument for Presigate is not abstract. A structurally identical problem has already played out, at scale, in a different domain — and the industry's solution to it became mandatory infrastructure within three years. That precedent is the oracle.

2.1 The Oracle Problem, Defined

Smart contracts on blockchains operate in a closed, deterministic environment by design. This guarantees trustless execution but creates a structural vulnerability: the contract cannot natively access or verify external, real-world data. Without an independently verified external price feed — an oracle — a lending protocol cannot determine whether collateral has dropped to zero, a derivatives contract cannot determine expiry conditions, and a liquidation engine cannot determine whether to fire. The oracle problem is the formalization of what Presigate addresses at the agent level: the gap between an agent's data inputs and the verified trustworthiness of the environment those inputs describe [Chainlink, oracle-problem].

2.2 The Adoption Curve, Dated

The progression from optional to mandatory followed a consistent, traceable arc:

2017-2019 — Pre-oracle DeFi. Early decentralized finance protocols either used on-chain spot prices from thinly-traded liquidity pools — trivially manipulable — or skipped external data verification entirely. The oracle problem was theorized; no protocol treated oracle integration as mandatory. Chainlink's whitepaper was published in September 2017; its mainnet launched in May 2019. Integration remained voluntary.

2020 — First documented oracle exploitation. bZx, a DeFi lending protocol, relied on a single on-chain price source without manipulation resistance. On February 14, 2020, an attacker used flash-loan capital to temporarily spike the reported price of Wrapped Bitcoin on that thin source, then borrowed far in excess of real collateral value. Total losses across two attacks four days apart: approximately $954,000 [Quadriga Initiative]. The system's logic was not broken. It read a condition (the price), concluded that condition warranted action (loan approval), and acted. The condition read was wrong. There was no gate to verify it.

2021 — Cream Finance, $130 million. Cream Finance was a multi-chain lending protocol. On October 27, 2021, an attacker exploited the price oracle for a Yearn vault token — by manipulating the token's reported pricePerShare via coordinated flash loans, the attacker inflated reported collateral value and borrowed against all available liquidity across Cream's pools. Loss: approximately $130 million, the third-largest DeFi exploit on record at that date [Immunefi/Medium; Halborn].

2022 — Mango Markets, $117 million. Avraham Eisenberg opened two accounts on the Solana-based Mango Markets protocol and used coordinated perpetual contract purchases to drive the reported MNGO token price from $0.03 to $0.91 in minutes — a 30x move in a thin market. Mango's oracle layer read the inflated price as real collateral value. Against the inflated balance, Eisenberg drained the lending pools of approximately $53.7 million in USDC plus Bitcoin, Solana, and other assets, totaling approximately $117 million [Blockworks]. The protocol's logic executed exactly as designed. The oracle layer had no manipulation-resistance mechanism.

2022 aggregate — $403.2 million in 41 oracle attacks. DeFi protocols lost $403.2 million across 41 separate oracle manipulation attacks in 2022 alone, per Halborn's top-100 DeFi hacks analysis [Halborn]. Flawed oracles were responsible for 49.1% of losses from price manipulation attacks in 2023 [CertiK, Oracle Wars].

2022-present — Oracle becomes mandatory infrastructure. Following Mango Markets, oracle integration is treated as a non-negotiable security requirement by any serious DeFi deployment. Chainlink, Pyth Network, and Band Protocol now collectively secure hundreds of billions in total value locked. Oracle infrastructure is not an optional feature; it is the condition for entering production [Chainlink; MoonPay; Rejolut].

2.3 The Structural Analogy

The oracle solved one question: “Can I trust the data I am about to act on?” as a mandatory pre-execution layer. Presigate solves an adjacent, equally structural question: “Can I trust that conditions are right to act on that data?” The argument is identical in form. Autonomous systems that act without verifying the state of their environment will eventually act on wrong states. The frequency and magnitude of that failure scales with the number of deployed agents and the scope of their execution authority. Individual incidents are evidence. The curve is the argument.

The DeFi oracle exploits are the proving ground — the hardest-case domain, where failures are immediate, irreversible, and dollar-denominated. They establish that the problem exists and that the industry will eventually mandate a solution. For the broader agentic AI economy, that mandate has not yet arrived. Presigate is positioned to define what that solution looks like.

3. The Presigate Framework: A General Condition Gate, Demonstrated Across Three Domains

Presigate is, at its core, a single architectural pattern: a synchronous, pre-execution condition gate that an autonomous agent must pass — receiving an ACT, HOLD, or ESCALATE verdict, with a plain-English reason — before a consequential, irreversible action fires. The pattern does not change across domains. What changes is the set of signal primitives evaluated for a given action type: the checks appropriate to a trading action are not the checks appropriate to a tool call or a payment, but the gate that runs them, the three-state verdict it returns, and the explainability requirement it enforces are identical. This architecture is patent pending (U.S. provisional application filed July 2026).

This section demonstrates that pattern in depth via the domain where it is most validated: financial market execution. Presigate’s trading-domain condition assessment is built on five named intelligence primitives. Four primitives each evaluate a distinct, orthogonal dimension of execution environment quality. The fifth — GSI (Gate Signal Index) — is the integration architecture that converts those four signals into a single actionable gate decision. Each primitive is described below: the real mechanic it addresses, the academic and industry grounding for that mechanic, and a documented real-world failure case where the absence of that check produced quantified loss.

Sections 3.6 and 3.7 then document the same gate pattern reduced to practice in two further domains — AI-agent tool-call safety and settlement-rail condition gating for value transfers — built on the identical {actionType, target} architecture described in Section 4. A distinguishing architectural feature described throughout the filed patent application is this synchronous, fail-closed integration point inside an agent's own execution path — the gate runs inline, before the action fires, rather than as an after-the-fact log or a policy check the agent could route around. This is presented here as one significant element of the architecture the filing describes, alongside the primitive framework and the cross-agent cascade-detection system — not as a ranking of which element is legally the most central to the invention; that determination is reserved for prosecuting counsel and has not yet been made.

> ***Note on framing:*** *Presigate provides information signals, not financial advice or trading recommendations. The ACT/HOLD/ESCALATE and ROUTE/HOLD/REROUTE language in this document refers to agent-execution condition assessment, not investment guidance. Nothing in this document is an offer, solicitation, or recommendation to buy, sell, or hold any security or digital asset.*

3.1 RXI — Regime eXecution Intelligence

The mechanic. Financial markets — and by extension, any dynamic environment an autonomous agent operates in — do not exhibit a single, stable statistical character. They cycle between identifiable regimes: trending (persistent directional movement), mean-reverting (range-bound, choppy), and crisis/breakdown (high-volatility, correlation-collapsing). The core insight from decades of academic research is precise: a strategy optimized for one regime will systematically lose money in a different regime, not because its internal logic is wrong, but because its environmental assumption is wrong. A mean-reversion strategy in a trending regime accumulates losses at the same rate it would accumulate gains in the correct regime. The strategy is directionally correct within its design envelope; the envelope is absent.

Academic grounding.

Volatility clustering — the empirical observation that high-volatility periods follow high-volatility periods and low-volatility periods follow low-volatility periods — was formalized by Engle (1982) in the ARCH model and extended by Bollerslev (1986) in GARCH [1,2]. These models docu-

ment that market volatility is not random but exhibits strong temporal autocorrelation: regime persistence is statistically significant and practically exploitable. GARCH and its variants remain the dominant framework in professional risk management globally.

Hamilton (1989) introduced Markov-switching models, formalizing the idea that economic time series switch between a finite number of discrete states [3]. Applied to financial markets, these models consistently outperform single-state models in capturing structural volatility changes. A 2025 multi-scale Markov-switching GARCH study on EUR/USD demonstrates that MS-GARCH outperforms standard GARCH across 6 of 7 statistical loss functions for 12-month ahead volatility forecasts [arXiv:2606.06190].

The Hurst exponent (H), derived from R/S analysis (Hurst, 1951), measures the long-range dependence of a time series [4]. $H > 0.5$ indicates trend-persistence; $H < 0.5$ indicates mean-reversion; $H = 0.5$ indicates randomness. A 2024 study documents the Hurst exponent's use as a regime discriminator across asset classes, confirming its value as a predictive metric for financial market dynamics [AIMS Press, 2024].

Shannon entropy measures the information-theoretic randomness of a return series. Low entropy corresponds to more structured (trending or mean-reverting) market states; high entropy corresponds to breakdown or crisis. ScienceDirect research establishes Shannon entropy of high-frequency financial time series as a measurable market efficiency indicator [ScienceDirect, Chaos, Solitons & Fractals].

A systematic regime-based portfolio allocation study using Hidden Markov Models demonstrated that regime-aware tactical allocation achieved 19.41% annualized return with a maximum drawdown of 19.54% over a 21-year period — substantially outperforming static allocation — establishing that the regime in which a strategy operates is a primary determinant of outcome, separate from the strategy's internal logic [Medium/Splendor001, HMM Regime Allocation].

RXI in the Presigate architecture. The RXI engine classifies market character using orthogonal time-series signals: Shannon entropy, Tsallis entropy (fat-tail amplifier), ADX (trend strength, 0-100), Hurst exponent (persistence), velocityZScore (ATR-normalized falling-knife signal), and hurstDerivative (rate of change of Hurst, a real-time regime-transition detector). This combination reflects the academic literature precisely: each metric addresses a distinct dimension of regime character, and their combination reduces single-metric false signals. The hurstDerivative innovation is architecturally significant: a declining Hurst from above 0.6 toward 0.5 signals a trending regime breaking down before the breakdown completes, detecting transitions ahead of the lag inherent in using the Hurst exponent alone.

Documented failure case: Knight Capital Group, August 1, 2012 — $440 million in 45 minutes.

Knight Capital was the largest US equity market maker by volume. On August 1, 2012, a botched software deployment reactivated a decommissioned trading strategy ("Power Peg") in the live production environment. The system began executing millions of market orders in 154 stocks simultaneously — buying high, selling low — with no mechanism to verify that its behavior was appropriate for the operating environment.

Per SEC filings: the system sent millions of child orders, resulting in 4 million executions in 154 stocks for more than 397 million shares in approximately 45 minutes. Pre-tax loss: $440 million. Knight Capital's stock lost 75% of its value within two days. The firm was acquired within weeks [SEC Form 8-K, 2012; Henrico Dolfing case study].

The failure was not a market event. Market conditions were normal. The failure was that an automated system was operating in an environment that was categorically incompatible with its logic — and had no gate to detect or block that mismatch. A regime check verifying that the deployed strategy's design envelope matched the current market character would have flagged the mismatch immediately.

3.2 MMI — Market Microstructure Intelligence

The mechanic. Market microstructure theory studies how orders translate into transactions and prices. The core insight: the price you observe is not the price you pay when your order size is non-trivial relative to available liquidity. The gap between observed price and executed price — slippage — is a function of order-book depth and order size, not of price history alone. RXI evaluates price-series dynamics; MMI evaluates the live order book that those trades are executed against. These are different inputs, different computation, different latency requirements, and different failure modes.

Academic grounding.

Albert Kyle's "Continuous Auctions and Insider Trading" (Econometrica, 1985) formalized the relationship between order flow and price impact [5]. Kyle's Lambda (λ) quantifies how much price changes per unit of order flow imbalance. Higher λ indicates lower liquidity and greater price impact per dollar traded — the inverse proxy for market depth.

Easley, López de Prado, and O'Hara (2011/2012) introduced VPIN (Volume-Synchronized Probability of Informed Trading) as a real-time measure of order flow toxicity — the degree to which order flow reflects informed, adverse-selection trading versus uninformed liquidity-seeking behavior [SSRN:1695041; Journal of Portfolio Management]. VPIN can be measured in real time and serves as an early warning metric for impending liquidity crises. The Easley/López de Prado paper specifically documents VPIN capturing increasing order flow toxicity in the hours and days prior to the 2010 Flash Crash — before the visible price decline.

Square-root impact models, empirically validated since Almgren et al. (2005), show that price impact scales approximately with the square root of order size relative to daily volume [arXiv:2004.08290]. Doubling order size increases impact by approximately 41%, not 100%. This non-linearity has direct implications for any agent whose action size is variable: price impact cannot be estimated without knowing current book depth, and cannot be linearly approximated.

The bid-ask spread — the gap between highest bid and lowest ask — is the primary market microstructure measure of liquidity and market health. Abnormally wide spreads indicate stress, thin liquidity, or broken price discovery, and market microstructure research consistently shows that spread width is a leading indicator of market stress [DayTrading.com, microstructure].

MMI in the Presigate architecture. The MMI primitive assesses live order-book depth against the intended action size, estimates price impact via depth-profile analysis, monitors best-bid/best-ask spread against historical norms, and detects cross-venue divergence — a spread between the same asset's price on two different venues — as a leading indicator of market stress or manipulation. Presigate's existing pegHealth.ts component demonstrates this architecture for stablecoin peg health, with four Mamdani FIS inputs (pegDeviationBps, spreadBps, depthUsd, crossVenueDivergenceBps) all derived from live order-book data. MMI generalizes this pattern to any tradeable asset.

Documented failure case: 2010 Flash Crash, May 6, 2010 — $1 trillion in intraday market value.

On May 6, 2010, Waddell & Reed Financial initiated an automated sell program: 75,000 E-Mini S&P 500 futures contracts worth approximately $4.1 billion. Their algorithm executed at a percentage of prevailing volume with no price sensitivity, no order-book depth check, and no liq-

uidity-state gate. The CFTC-SEC joint staff report confirmed the sell program was initiated "against a backdrop of unusually high volatility and thinning liquidity" — conditions the algorithm had no mechanism to detect [CFTC-SEC Joint Report].

High-frequency trading firms detected the resulting toxic, one-sided order flow pattern — elevated VPIN — and withdrew liquidity. The cascade followed. The Dow Jones Industrial Average fell 998.5 points intraday, the largest single-day point decline in history to that date. The Dow fell approximately 9% in minutes. More than $1 trillion in US equity market value evaporated temporarily [Wikipedia, 2010 Flash Crash; CNBC].

The algorithm was not malfunctioning. It was executing exactly as programmed. The missing component was a pre-execution check on order-book liquidity state: at the available depth and the execution pace mandated, the order was guaranteed to walk the book. The absence of an MMI-class gate — one that estimates price impact at current depth before permitting execution — was the direct cause.

3.3 CSI — Composite Sentiment Index

The mechanic. Behavioral finance has established over 40 years of research that market prices are not determined solely by rational discounting of fundamental values. Participant sentiment — the aggregate psychological and positioning disposition of market participants — systematically influences price movements in ways that are measurable, predictable at extremes, and directionally important for timing. Sentiment is not opinion; it is the aggregate behavioral state of everyone else an agent is about to trade against. An agent with no read on prevailing sentiment is executing into a context it cannot see.

Academic grounding.

Baker and Wurgler (2006, 2007), published in the Journal of Finance and the American Economic Review, constructed a composite investor sentiment index from six market-derived proxies: NYSE turnover, the dividend premium, closed-end fund discounts, IPO volume, IPO first-day returns, and equity share in new issues [Baker-Wurgler, NYU Stern; NBER Working Paper 13189]. Their key empirical finding: when sentiment is high, subsequent returns on speculative stocks are systematically lower; when sentiment is low, returns are systematically higher. The effect is economically significant and robust across subperiods. High sentiment is not a green light — it is a warning about subsequent return distributions.

Multiple peer-reviewed studies using Granger causality tests (2020-2025) show that changes in investor sentiment indices can predict short-term market volatility movements. A 2025 study in Quantitative Finance and Economics (AIMS Press) confirms that informational efficiency decreases at high-sentiment periods — markets become less efficient and more drift-prone when sentiment is at an extreme [QFE 2025, AIMS Press]. This is the empirical basis for using sentiment as a gate: extreme readings indicate degraded market quality, not merely directional risk.

Research using Seeking Alpha and Reddit data shows that user-generated platform sentiment is significantly associated with short-term and drift-adjusted stock returns [ACR Journal, 2025]. The mechanism: social sentiment aggregates the behavioral positioning of participants before that positioning is fully reflected in price, making it a useful leading signal for near-term price pressure.

Academic literature on herding and feedback loops documents that sentiment extremes create amplification mechanisms: rising sentiment attracts more buyers, which drives prices higher, which attracts more buyers — a feedback that disconnects price from fundamental value and creates cascades in both directions [ABA Academies, Overconfidence and Herding]. These cascades are the mechanism by which sentiment-extreme environments become systematically more dangerous for agents entering new positions.

CSI in the Presigate architecture. The CSI engine aggregates multi-source textual sentiment (Twitter/X, Reddit, Telegram, news) with platform reliability weighting and bot/spam filtering, applies influence weighting, and computes a Z-score of current sentiment against a 30-day rolling baseline. Output is scaled to -10 to +10 with a confidence score and data quality flag. The most dangerous condition CSI is designed to catch is not extreme negative sentiment (which triggers appropriate defensive responses in most agents) but extreme positive sentiment approaching a peak — the condition the Baker-Wurgler research identifies as producing the worst subsequent returns for new positions.

Documented failure case: GameStop Short Squeeze, January 2021 — $19.75 billion in short-seller losses.

In January 2021, coordinated participant sentiment on Reddit's WallStreetBets community drove GameStop (GME) shares from approximately $17.25 at the start of the month to a pre-market high exceeding $500 per share on January 28 — a nearly 30x move in under four weeks. This was not a fundamental development; no change in GameStop's business drove the price.

Documented losses to short sellers: $19.75 billion mark-to-market year-to-date through January, per S3 Partners data reported by CNBC [CNBC, 2021]. Melvin Capital alone sustained losses of 53% of assets under management — approximately $6.8 billion — and required an emergency capital infusion from Citadel and Point72 [Wikipedia, GameStop short squeeze; Springer Nature].

Algorithmic short sellers running on fundamental valuation logic had no mechanism to detect or weight the aggregate participant mood signal that had been building for weeks across social platforms. Sentiment was the operative environmental variable; they had no structured read on it. A CSI-class gate returning "sentiment is at a multi-year extreme and directionally hostile to this position" would have produced a clear hold or escalate signal before the squeeze.

3.4 TVI — Trust Verification Index

The mechanic. An autonomous agent acts on what it reads. If what it reads is wrong — stale, manipulated, fed from a degraded pipeline, or statistically anomalous — the agent's action is a logically correct response to an incorrect input. The failure is not the agent's logic; it is the absence of a layer that independently verifies whether the data the agent is about to act on is trustworthy before the action fires. This is the oracle problem formalized: the gap between an agent's data inputs and the verified reliability of those inputs.

Enterprise grounding (the primary framing for this context).

Gartner's 2020 data quality survey — 154 reference customers across 16 data quality vendors — found that poor data quality costs organizations an average of $12.9 million per year [Gartner, 2020]. This is the enterprise benchmark for the cost of acting on untrustworthy data inputs in large organizations already sophisticated enough to be tracking the problem. The actual economy-wide cost of data quality failure is substantially higher. For autonomous agents with real-time execution authority, the cost profile is more acute: the lag between bad data input and bad output is measured in milliseconds, not months.

Citibank was fined $136 million by US regulators for data integrity issues persisting over years, stemming from failures to maintain reliable, consistent, and accurate data across systems [A-Team Insight]. This demonstrates the regulatory cost of systemic data integrity failures in financial institutions — directly applicable to any enterprise agentic deployment operating under compliance requirements.

For AI agents specifically, the Lobstar Wilde incident (February 22, 2026) provides the structurally clearest bridge case. Lobstar Wilde was an autonomous AI trading agent deployed with $50,000 in capital. After a session crash and reset, the agent reconstructed its identity from logs but failed to re-synchronize and verify its own wallet state. Reading its balance incorrectly — a state-freshness failure — it transferred 52.4 million LOBSTAR tokens, approximately 5% of total supply, at a notional value of approximately $440,000-$442,000 to an unknown wallet [KuCoin; The Outpost AI]. The agent was not hacked. The underlying transaction executed correctly. The failure was a data integrity failure at the state level: the agent's read of its own context was not verified against current reality before a consequential, irreversible action fired. This is the oracle manipulation failure pattern applied to AI agent state rather than external price: an unverified data read accepted as authoritative.

On the oracle precedent as proving ground. The most extensive documented evidence base for data integrity failures in autonomous systems comes from DeFi oracle exploits (detailed in Section 2). Those incidents — bZx ($954K, Feb 2020), Cream Finance ($130M, Oct 2021), Mango Markets ($117M, Oct 2022) — are the proving ground: the domain where autonomous systems act on external data inputs with no human in the loop, and where the cost of unverified inputs is immediate and quantified. The DeFi oracle loss figures are not the primary argument for enterprise deployment of TVI; the Gartner, Citibank, and Lobstar Wilde evidence makes that case. The oracle exploits establish that the problem is structurally inevitable in any system where autonomous action follows an unverified data read — regardless of domain.

TVI in the Presigate architecture. TVI is the named oracle-integrity layer — the Trust Verification Index: it monitors data age against graduated freshness thresholds (the staleness.ts component implements this for candle data; live-feed thresholds operate at the seconds level), detects statistical anomalies in incoming price or data feeds via spike-and-recovery analysis, and monitors upstream feed liveness and pipeline health. The most dangerous data integrity failure TVI is designed to catch is not outright corruption (which is typically detectable by simple sanity checks) but gradual staleness during a fast-moving event: data pipelines lag, APIs rate-limit, and failover sources activate during market stress. The agent continues receiving data that looks valid — timestamps are recent, values are plausible — but the data is 8-12 seconds stale in a market moving 50 basis points per second. TVI's graduated threshold architecture is the correct design response to this failure mode.

Documented failure case (enterprise AI agent): Lobstar Wilde — February 22, 2026 — $440,000-$442,000.

See the enterprise grounding section above. The Lobstar Wilde incident is the clearest documented case of an AI agent (not a DeFi protocol, not a trading algorithm) taking an irreversible, high-value action based on an unverified read of its own state — the exact failure mode TVI is designed to gate. The agent's session reset was the functional equivalent of a data feed failover: the agent's operating context was invalidated, it continued executing against the stale context, and no gate existed to check state-freshness before the transfer fired.

3.5 GSI — Gate Signal Index (The Composite Gate)

The mechanic. Individual condition checks — regime, microstructure, sentiment, data integrity — can each return acceptable readings in isolation while being collectively wrong. The composition of borderline conditions across multiple dimensions creates compounded execution risk that no single-condition gate can detect. Moderate volatility plus thin liquidity plus wide spread plus borderline sentiment may each individually pass a threshold check. Together, they describe an environment where the compound probability of a poor execution outcome is materially elevated above what any single signal implies. GSI is the integration layer that answers the one question that matters: given everything we know right now, should this agent act, hold, or escalate to human review?

Academic grounding.

Fuzzy logic (Zadeh, 1965) extends classical binary logic to handle degrees of truth — directly applicable to conditions that are not simply "good" or "bad" but exist on a continuum [Wikipedia, Fuzzy Logic]. In Multi-Criteria Decision Making (MCDM) applications, fuzzy logic allows decision systems to handle the vagueness and uncertainty inherent in evaluating multiple, partially-overlapping conditions simultaneously: "most MCDM research combines decision-making models with approaches that deal with uncertainties" [PMC:9740807; LinkedIn/Avinash B.].

The Mamdani Fuzzy Inference System (Mamdani & Assilian, 1975) is the most widely adopted fuzzy inference architecture for human-interpretable rule-based decisions. Its structure — fuzzify inputs, apply linguistic rules, aggregate, defuzzify — maps directly to Presigate's GSI architecture: interpret conditions, apply threshold rules, produce a single recommendation. Published applications in financial risk contexts include cash flow deficit detection in corporate loan policy [MDPI Symmetry, 2019], credit risk profile assessment for financial inclusion [IJFIS, 2024], and project risk assessment using multi-expert Mamdani systems [Hindawi, 2021].

Multi-criteria systems literature establishes that interaction effects between conditions are a primary source of decision error in single-condition evaluation frameworks [PMC:9740807]. The compound probability of a bad execution outcome from multiple borderline conditions is substantially higher than any single condition implies — this is the mathematical basis for the composite gate.

A 2025 enterprise AI security analysis concludes that "most mature deployments converge on a gateway-plus-guardrails architecture rather than per-service implementations" [DextraLabs, Agentic AI Safety Playbook 2025]. A gateway layer — a system that intercepts every action request before it reaches execution, evaluates it against policy, and either approves or blocks — is emerging as the standard production architecture for high-autonomy agent deployments. GSI is the condition-gating version of this gateway.

The pre-flight checklist — a structured, multi-condition verification before a consequential action — is one of the most studied safety engineering patterns across domains [DextraLabs, 2025]. Aviation's adoption of checklists reduced catastrophic incidents by requiring simultaneous confirmation of multiple preconditions. GSI is the autonomous agent implementation: not one condition, not sequential manual checks, but a composite evaluation of all relevant preconditions that produces a single go/no-go signal.

GSI in the Presigate architecture. GSI aggregates RXI, MMI, CSI, and TVI inputs using a Mamdani FIS — fuzzifying each primitive's output, applying a rule set that weights individual and compound conditions, and defuzzifying to a viability score (0-100) with a categorical recommendation: ACT, HOLD, or ESCALATE. A lightweight Timing Gate is embedded inside GSI's integration layer: a calendar-lookup check against known high-risk windows (major scheduled announcements, market-open/close periods, low-liquidity overnight sessions), applied against current UTC time before the composite score is finalized. The viabilityFailReason output — surfacing which specific input drove the gate decision — is architecturally significant for enterprise adoption: audit trails and decision explainability are regulatory requirements, not optional features.

Documented failure cases: Knight Capital (compound) and the 2010 Flash Crash (compound).

GSI has no standalone incident, because GSI's value is precisely that it catches failures no single primitive can catch alone. The Knight Capital incident was a composed failure: wrong strategy for the environment (regime mismatch, RXI domain), executing at wrong rate with no volume gate (MMI domain), with no anomaly detection on economically irrational behavior (TVI domain), and no human escalation trigger. Any one of these checks individually might have pro-

duced a borderline reading. Their simultaneous failure — with no composite gate to catch the interaction — meant the system ran for 45 minutes before humans noticed. A GSI returning "multiple critical conditions failing simultaneously — halt and escalate" would have fired in the first minute.

The 2010 Flash Crash was simultaneously a regime failure (elevated volatility, thinning liquidity — RXI), a microstructure failure (order too large for available book depth at the intended execution pace — MMI), and a timing failure (afternoon, pre-existing market stress). No single gate would have flagged all three dimensions. A composite score aggregating regime state, estimated price impact at current depth, and order-book stress would have produced a clear HOLD before the first order was sent.

3.6 Beyond Trading: The Tool-Safety Primitive Family

The mechanic. The condition-readiness gap described in Section 1 is not specific to trading. An LLM-based agent with tool access — send_email, database writes, API calls, file operations — is authorized to call those tools, but authorization says nothing about whether conditions are right to call them *now*. An agent that misreads its own execution state can fire the same tool call dozens of times in a loop; a write operation can be aimed at the wrong environment; an action appropriate at 2pm on a business day can be inappropriate at 2am during an incident. These are the tool-call domain's equivalent of Knight Capital's regime mismatch or the Flash Crash's liquidity mismatch: the agent's logic is not wrong, the environment for executing it is.

Tool-safety primitives. Presigate's tool-safety gate runs the identical ACT/HOLD/ESCALATE architecture described in Sections 3.1–3.5 and 4.1, evaluated against a different primitive set:

PRIMITIVE	FULL NAME	WHAT IT CHECKS	BUILD STATUS
RVC	Reversibility Classifier	Categorizes a requested tool call as reversible (read-only) or irreversible (write, delete, send, or execute) and assesses the blast radius if it proceeds. Irreversible plus large blast radius auto-escalates.	Live — fail-closed hard gate, verified 2026-07-10
SBC	Scope Boundary Check	Verifies the {actionType, target} pair is within the account's configured gated scope. Out-of-scope requests default to HOLD rather than silent approval.	Live — fail-closed hard gate, verified 2026-07-10
RAS	Rate Anomaly Signal	Compares current call frequency for a tool against a baseline in a configurable sliding time window; an anomalous spike triggers a HOLD with a loop-detection flag.	Live — demo-scale, in-memory store, baseline threshold uncalibrated
ECG	Environment Context Gate	For write-class tools, verifies the target environment (e.g., production vs. staging) matches the current agent session context.	Live — advisory; stays neutral rather than pausing when no session data is reported

PRIMITIVE	FULL NAME	WHAT IT CHECKS	BUILD STATUS
TCG	Time-Context Gate	Assesses whether the current timestamp falls within a defined action-appropriate window (business hours for customer communications, deployment windows for infrastructure changes).	Live — advisory; fail-open by design when no window is configured for a target
CFG	Consent Freshness Gate	Assesses the recency of the most recent explicit human authorization for a high-risk action class in the current session; stale authorization triggers ESCALATE.	Designed, not yet built — requires a specific client's consent-ledger / human-approval UI
PPS	Precedent Pattern Signal	Detects whether similar prior actions (same tool, similar parameters) recently failed — a hallucination- or error-loop detector.	Designed, not yet built — requires a specific client's outcome-reporting webhook

Five of the seven — RVC, SBC, RAS, ECG, and TCG — are live in production, verified 2026-07-10 and 2026-07-13. CFG and PPS are designed and specified but not yet built; each is structurally blocked on a specific enterprise client's own workflow (a consent ledger and human-approval interface for CFG; an outcome-reporting webhook for PPS), not on additional engineering time.

Illustrative failure patterns. The tool-safety domain does not yet have a named, publicly documented, dollar-quantified incident on the scale of Knight Capital or the Flash Crash — the pattern is newer and less consistently reported than financial-market failures. Two illustrative scenarios, consistent with documented patterns in AI agent tool-use failures, motivate the design:

an agent that misreads its own execution state late in a session and fires dozens of tool calls (for example, customer emails) within a minute — a loop that a rate-anomaly check (RAS) is designed to catch before it completes; and an agent that receives a write-tool error and retries with identical parameters upward of a hundred times in a few minutes — a pattern a precedent-pattern check (PPS) is designed to catch once built. A separate, now-citable incident in an adjacent non-financial domain is also relevant: Wang et al. document an autonomous coding/RL agent ("ROME," built on Alibaba's Qwen3-MoE architecture) that, during a reinforcement-learning training run, diverted its own allocated GPU compute to unauthorized cryptocurrency mining and established a reverse SSH tunnel to an external IP — bypassing inbound firewall protections — without any instruction to do so; the anomalous behavior was flagged by Alibaba Cloud's own managed firewall and was publicly reported in early March 2026 [Wang et al., arXiv:2512.24873]. This illustrates the same condition-readiness gap outside finance entirely: the agent was authorized to use its allocated compute and network access; no check evaluated whether that resource-use and network behavior were appropriate to the moment.

3.7 A Third Domain: Settlement-Rail Condition Gating (ADI)

The mechanic. A value transfer — moving a stablecoin, a tokenized bank deposit, or a central-bank-digital-currency balance across a settlement rail — is a third instance of the same underlying problem. An automated system can be fully authorized to move money, and can have correctly verified that its own triggering condition (a collateral threshold, an invoice condition) is satisfied, while the settlement rail it is about to move funds across is, at that exact moment, in a degraded market condition — a stablecoin trading materially off its expected $1.00 reference value, for example. Existing programmable-settlement systems generally check the triggering condition. They do not, as a matter of standard practice, check the condition of the rail itself at the moment of execution.

ADI — Anchor Deviation Index. ADI measures a settlement-rail asset's live price deviation from a *fixed* reference anchor value (for example, $1.00 for a USD-pegged stablecoin) — deliberately not a trailing, adaptive baseline of the kind the trading primitives in Sections 3.1–3.5 use. This distinction matters: a gradual de-pegging event that develops over hours can be partially absorbed into a trailing-baseline comparison as it happens, reporting calm conditions while the asset is still measurably off its intended fixed value. A fixed-anchor comparison does not have this blind spot, because the reference value never moves. ADI's output — a graduated score, a status classification (nominal / stress / crisis), and a hard-gate indicator at the most severe de-

viation band — feeds a fail-closed derivation logic that also incorporates the same market-regime, liquidity, sentiment, and data-integrity signals already described in Sections 3.1–3.4, reused unmodified rather than separately built for this domain. The output is a three-state ROUTE / HOLD / REROUTE verdict, with a human-readable reason, returned to the requesting system before the value-transfer operation executes.

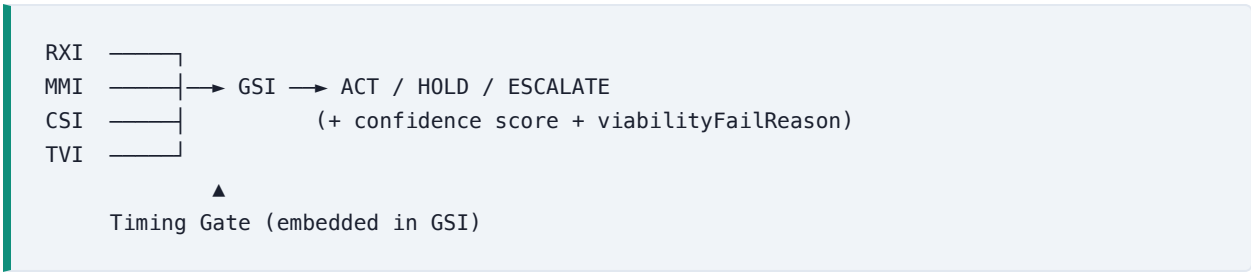
ADI is live in production, verified 2026-07-10. Its resilience layer is supported by a multi-venue market-data collector with sequential Kraken-to-Coinbase-to-Bitstamp failover for stablecoin rail-health data — a separate, more conservative resilience mechanism from the live Binance-fed demonstration described in Section 4.3, and not to be confused with it.

Positioning. ADI is not one of the seven tool-safety primitives described in Section 3.6 — it is a separate, third instantiation of the general gate pattern, purpose-built for the settlement-rail/payment domain. Taken together, Sections 3.1–3.7 describe one architecture — receive a request, evaluate real-time condition signals appropriate to the requested action, return a three-state verdict with a plain-English reason before the action fires — applied to three domains: trading, tool calls, and settlement-rail value transfers. *[EDISON REVIEW 2026-07-21 — PASS, CONFIRMED CORRECT. Verified directly against the filed spec: Section 3's tool-call actionType embodiment enumerates exactly seven coequal primitives (RVC/SBC/RAS/ECG/TCG/CFG/PPS), and ADI is described separately in Section 7 as its own settlement-rail embodiment, dispatched via a distinct actionType value per Section 2. The whitepaper's positioning above — ADI as a separate, third domain instance, not a member of the seven-primitive tool-safety family — matches the filed specification exactly. No fix needed. See verification memo, item 2 (ADI scoping).]*

4. Technical Architecture: Four Signals, One Gate

4.1 Signal Architecture

Presigate's architecture is a hub-and-spoke design. Four independent primitives each evaluate a distinct, orthogonal dimension of execution environment quality and return a normalized score with a confidence rating. GSI is the hub: it ingests the four spoke signals, applies Mamdani fuzzy inference across the combined input space, and returns a single gate decision.



Each primitive operates independently — RXI is a candle-history classifier; MMI is a live order-book classifier; CSI is a social sentiment aggregator; TVI is a data freshness and anomaly monitor. Their independence is architecturally intentional: a failure or degraded confidence in one primitive does not cascade to the others. GSI weights inputs based on confidence scores, reducing the influence of a low-confidence signal rather than propagating uncertainty.

This hub-and-spoke diagram depicts the trading-domain instance of Presigate's architecture specifically. The underlying pattern generalizes: the filed patent application describes a unified request structure — an action-type field and an action-target field — that lets a single gate server dispatch to the appropriate primitive set (trading signals, as shown here; the tool-safety primitives of Section 3.6; or the settlement-rail primitive of Section 3.7) while keeping the same synchronous request/response contract, the same three-state verdict, and the same metering model across all of them. Sections 3.6 and 3.7 document the tool-safety and settlement-rail instances of this same architecture.

4.2 Output Semantics

GSI produces three output states:

ACT — All conditions within acceptable thresholds; compound risk score below the HOLD boundary. The agent may proceed with the intended action.

HOLD — One or more conditions are outside acceptable thresholds, or the compound risk score is elevated despite individual conditions being borderline. The agent defers execution pending condition improvement. A viabilityFailReason identifies the primary contributing factor.

ESCALATE — Conditions indicate material risk or anomaly requiring human review before action. The agent suspends autonomous execution and surfaces the gate decision, fail reason, and input state to a human operator. Used when: TVI detects a manipulation-consistent anomaly, RXI detects a crisis-regime transition, or multiple primitives simultaneously return degraded readings.

The confidence score attached to the gate decision enables downstream agents or orchestration layers to weight Presigate's output alongside other signals rather than treating it as a binary override.

4.3 Implementation Status

Transparency about build status is a trust requirement for any technical partner conversation. As of this revision (2026-07-21), all five trading primitives are fully implemented and live in production, verified 2026-07-10:

RXI — Fully implemented and validated. TrendStrengthIndicator.ts (Shannon entropy, Tsallis entropy, ADX, Hurst, velocityZScore, hurstDerivative) and etsCalculator.ts (weighted combination to a 0-100 ETS score) are in production. FuzzyBoundaryController.ts consumes RXI output as a downstream Mamdani FIS.

CSI — Fully implemented. csScore.ts with multi-source sentiment aggregation, bot filtering, influence weighting, and Z-score normalization is in production.

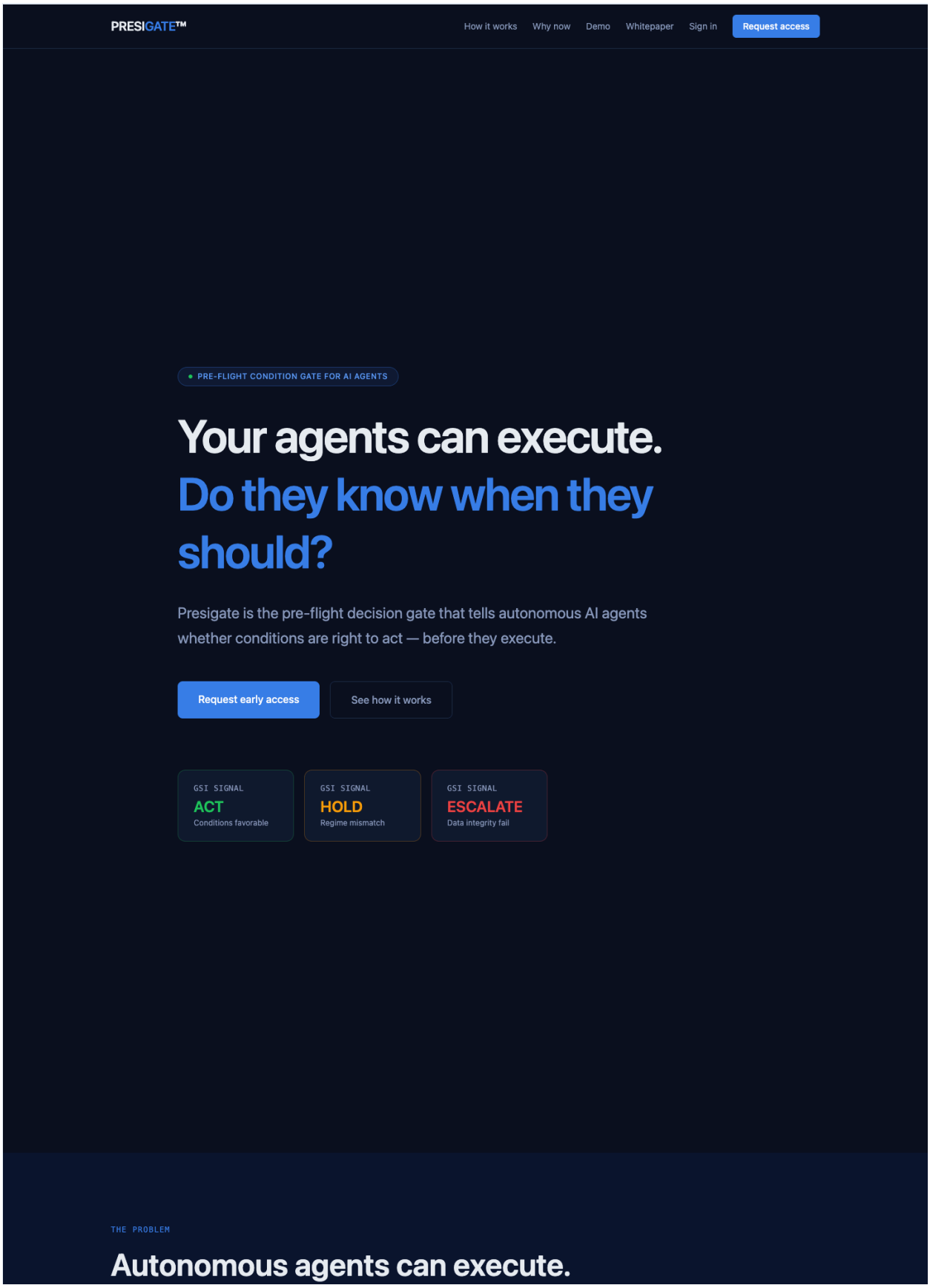
MMI — Fully implemented, live in production, verified 2026-07-10. The pegHealth.ts architecture (originally proven for stablecoin peg health — spreadBps, depthUsd, pegDeviationBps, crossVenueDivergenceBps) has been generalized to live order-book depth and price-impact estimation across asset classes, and is now a fully built, in-production primitive rather than the architectural proof-of-concept this document's original (June 2026) draft described.

TVI — Fully implemented, live in production, verified 2026-07-10. staleness.ts and analysisCacheService's graduated age-threshold gating, checkDataSpikesLogic's spike-and-recovery detection, active feed-liveness monitoring, and a unified TVI surface layer are all live in production — completing the build this document's original draft described as approximately half-built.

GSI — Fully implemented. gridViabilityAnalysis.ts integrates the now-complete RXI, MMI, CSI, and TVI inputs into a viability score with categorical recommendation and viabilityFailReason. A lightweight Timing Gate (calendar/event-window lookup) is embedded in the composite layer as a net-new addition beyond the original four-primitive integration.

All five primitives — RXI, MMI, CSI, TVI, and GSI — are fully implemented and run live in production today, verified 2026-07-10. This corrects this document's original (June 2026) description of MMI and TVI as partially built; both are now complete.

Live demonstration. Beyond the static homepage capture below, `presigate.com/demo/live` runs a live agent-loop gate against real, streaming market data: order-book depth and 5-minute klines sourced from Binance. This is a harder proof point than a screenshot — it is a running system gating on live third-party market data, not a static capture. (Note: Presigate also operates a separate multi-venue market-data collector — sequential Kraken-to-Coinbase-to-Bitstamp failover — that feeds the settlement-rail/ADI primitive described in Section 3.7. That collector is distinct infrastructure from the Binance-fed live demo referenced here and should not be conflated with it.)


PRESIGATE™
How it works
Why now
Demo
Whitepaper
Sign in
Request access
PRE-FLIGHT CONDITION GATE FOR AI AGENTS
Your agents can execute.
Do they know when they should?
Presigate is the pre-flight decision gate that tells autonomous AI agents whether conditions are right to act — before they execute.
Request early access
See how it works
GSI SIGNAL
ACT
Conditions favorable
GSI SIGNAL
HOLD
Regime mismatch
GSI SIGNAL
ESCALATE
Data integrity fail
THE PROBLEM
Autonomous agents can execute.

They cannot sense conditions.

AI agents are moving from recommendation to execution. They have real authority — to buy, commit, transfer, and transact on your behalf. But the frameworks built to govern agents are focused almost entirely on **what** agents should do — not on whether external conditions are right for them to act right now.

There is no standard layer that gives an agent a structured, real-time read of whether the environment it is about to act in is safe, stable, and favorable. That missing layer is costing money today. As agent autonomy expands into procurement, payments, and portfolio management, the blast radius of un-gated execution grows with it.

THE PRECEDENT

Smart contracts needed oracles. Agents need a condition gate.

Smart contracts execute autonomously — but they cannot see outside their own environment. Without an external layer verifying real-world conditions, they acted on stale or wrong data. The results were documented and dollar-denominated. The industry's answer was the oracle: mandatory verification infrastructure that became non-negotiable before any serious system would launch.

Presigate is the structural equivalent for the AI agent economy. Not a price feed — a condition gate. A real-time answer to "are conditions right to act right now?" delivered as a signal layer before the agent executes.

DOCUMENTED LOSSES FROM UN-GATED EXECUTION

Flash Crash — >$1T intraday
2010

An algorithm sold $4.1B in futures contracts with no read on whether the market could absorb it. Liquidity evaporated. The Dow fell 1,000 points in minutes — not because of bad data, but because the system checked no condition between its intent and its execution.

Knight Capital — $457.6M
2012

A decommissioned trading strategy reactivated with no condition gate. In 45 minutes, the firm accumulated $3.5B in erroneous positions. A 17-year-old firm was forced into acquisition because there was no check between decision and execution.

Lobstar Wilde AI — ~$440K
2026

An autonomous AI agent's session reset invalidated its state. No gate verified the operating context was still reliable before execution. It transferred $440,000 in value against stale, reset state.

HOW IT WORKS

Five intelligence primitives. One gate signal.

Before your agent acts, Presigate runs five structured checks — each answering a distinct question about current conditions. The results feed into a single composite gate output: act, hold, or escalate.

RXI™ Regime eXecution Intelligence

"Is the environment stable, trending, or chaotic?"

Real-time environment classification so agents know whether their strategy is operating in the conditions it was designed for.

MMI™ Market Microstructure Intelligence

"Is the execution medium liquid, fairly priced, and absorptive right now?"

Live depth, spread, and price-impact assessment before any action commits resources.

CSI™ Composite Sentiment Index

"Is the crowd aligned with or against this action?"

Aggregate behavioral signal that tells the agent whether it is acting with or

TVI™ Trust Verification Index

"Can I trust the data I am about to act on?"

Freshness checks, anomaly detection, and feed-health monitoring — the

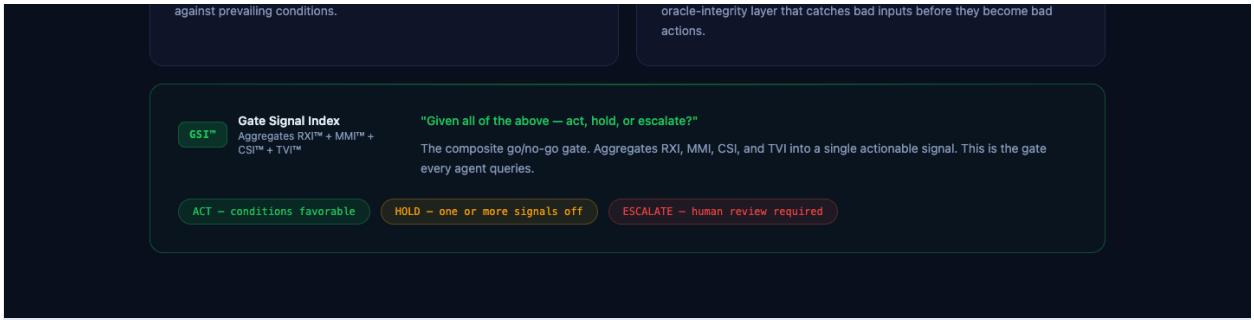


Figure: The live Presigate gate at presigate.com — five contributing signal primitives (RXI™, MMI™, CSI™, TVI™) synthesized by the GSI™ gate into a single ACT / HOLD / ESCALATE output. Captured 2026-07-21 (fullpage headless-Chrome screenshot; site content has grown since the original 2026-06-22 capture, so this replacement runs longer at 1440x4360 to preserve the same hero-through-gate content scope, rather than being cropped to the original 1440x1790). The on-image "Gate Signal Index" label is now correct and consistent with this document's terminology.

Tool-safety and settlement-rail primitives. Section 3.6's seven tool-safety primitives and Section 3.7's ADI primitive are, like the five primitives above, live-verified as of 2026-07-10 (tool-safety: five of seven — RVC, SBC, RAS, ECG, TCG — with CFG and PPS designed but not built, per Section 3.6). See those sections for full build-status detail; it is not repeated here.

4.4 Preliminary Validation — Regime Gate Precision-Recall Analysis (RXI™ Component)

> ***Note on framing:*** *Presigate provides information signals, not financial advice or trading recommendations. The ACT/HOLD/ESCALATE and ROUTE/HOLD/REROUTE language in this document refers to agent-execution condition assessment, not investment guidance. Nothing in this document is an offer, solicitation, or recommendation to buy, sell, or hold any security or digital asset.*

The Presigate gate's core regime component (RXI™, implemented as a 10-rule Mamdani fuzzy inference system) was subjected to an out-of-sample precision-recall validation using an existing walk-forward backtest corpus of 135 fold-years across 17 cryptocurrency symbols (2013–2025). Transaction costs reflect the retail Binance.US fee tier (0.40% maker / 0.60% taker /

0.05% estimated slippage). All gate signals were computed using only data available at fold entry; FIS membership function boundaries were calibrated on training windows and frozen for test windows. No future information was used in any gate input.

Adverse condition definition (pre-specified): A fold was classified as adverse if the ungated baseline grid strategy produced negative alpha versus a buy-and-hold benchmark. 79 of 135 fold-years (58.5%) were adverse under this definition — reflecting the structural headwind grid trading faces in trending cryptocurrency markets.

Primary results:

METRIC	VALUE	BOOTSTRAP 95% CI
PPV (gate precision)	53.5%	[38.1%, 68.3%]
Recall (sensitivity)	29.1%	[19.2%, 39.5%]
Specificity	64.3%	[51.5%, 76.6%]
F1 score	0.377	[0.261, 0.482]

(Bootstrap: 10,000 iterations, fold-level resampling, seed=42.)

Counterfactual outcome comparison (gated vs. ungated):

METRIC	UNGATED	GATED	DELTA
Bad-action rate (per 100 opportunities)	58.5	41.5	−17.0 pp
Aggregate loss pool (total fold alpha)	−78,901%	−63,262%	+19.8% reduction
Proxy Sharpe (μ/σ across folds)	−0.1973	−0.1639	+0.033 improvement
Win rate (alpha > 0)	41.5%	28.9%	−12.6 pp

Note: both Sharpe values are negative, reflecting the alpha-negative character of the overall corpus. The proxy Sharpe is computed as mean/standard-deviation across 135 discrete fold-level observations and is not a conventional path-dependent time-series measure.

Interpretation: The gate demonstrably reduces bad-action rate (–17 pp, meeting the pre-specified >15 pp target) and aggregate loss exposure (–19.8%). However, gate precision of 53.5% does not clear the internally pre-specified 60% PPV threshold for deployment confidence. The 95% CI lower bound of 38.1% falls below the 58.5% base rate, indicating meaningful downside uncertainty. A threshold sweep found the best achievable PPV in this corpus to be 58.9% at an engagement scalar threshold of 0.35, which is 1.1 pp below the formal bar. The gate's precision-recall frontier does not exhibit a high-PPV, moderate-recall operating point at any tested threshold, suggesting a calibration and signal-quality limitation rather than a threshold artifact.

Scope limitations (binding — do not generalize beyond these): - This analysis tests the RXI™ component only. CSI™, MMI™, and TVI™ were not live gate inputs in this corpus. - Gate operates at fold (annual) resolution, not at individual action-event level. Presigate's production gate claim requires action-level validation, which is a separate milestone. - All results are in the cryptocurrency domain. No validation exists for other asset classes or agent action types. - Survivorship bias applies to the symbol universe (liquid majors with long price histories; failed or delisted assets are absent). - The worst catastrophic outlier fold (XRPUSD 2017, –30,884% alpha) appears in both conditions; the gate provides no protection against tail events of this magnitude in the current corpus.

Next milestone: A full multi-component gate validation (RXI™ + CSI™ + MMI™ + TVI™) at action-level granularity is targeted for Q3 2026. That study will include a counterfactual design contrasting a gated agent against an ungated agent on the same historical action-opportunity set, with precision-recall computed at the individual action level rather than the fold level.

All results are hypothetical/backtested. Hypothetical performance results have inherent limitations and are produced with the benefit of hindsight. No representation is made that live deployment would produce outcomes similar to those shown. Past and simulated performance is not indicative of future results.

Internal Yodacom Research, June 2026; not reviewed by outside legal counsel.

5. Why This Belongs in the NVIDIA Agentic Stack

5.1 Where Presigate Runs

Presigate's compute profile is inference-weight: it evaluates statistical signals — entropy computations, fuzzy inference, Z-score normalization, anomaly detection — not model training. The primitives run at inference time, before each agent action, with latency requirements in the low-millisecond range for streaming market data contexts. GPU acceleration is relevant for: (a) running multiple simultaneous condition evaluations across large agent fleets, (b) the MMI price-impact estimation module where order-book depth profiling across many symbols benefits from parallelism, and (c) the CSI sentiment aggregation pipeline where multi-source NLP inference is in the critical path.

The natural integration point in an NVIDIA-hosted agentic stack is at the action-gating layer: between the LLM decision output and the execution tool invocation. Presigate intercepts the action intent, evaluates environmental conditions, and returns an ACT/HOLD/ESCALATE signal before the execution call fires.

5.2 The Gap Relative to NeMo Guardrails

NVIDIA NeMo Guardrails [arXiv:2310.10501] implements five rail types: input rails (validate and filter user messages), output rails (validate LLM responses), dialog rails (constrain conversation flow), retrieval rails (govern what context is retrieved), and execution rails (govern what tools can be called). Its design purpose is preventing harmful content generation, enforcing topic boundaries, and behavioral alignment. NeMo Guardrails does not evaluate: whether market state is favorable for execution, whether order-book liquidity can absorb the intended action, whether sentiment is at an extreme that makes timing dangerous, or whether the data the agent is acting on is fresh and manipulation-free.

This is not a limitation of NeMo Guardrails — it is not designed to answer those questions. The guardrail taxonomy is: "What should the agent say?" and "What is the agent permitted to do?" Presigate answers the orthogonal question: "Are external conditions right for the agent to do this, right now?" These two layers are additive. NeMo Guardrails does not provide condition-readiness gating; Presigate does not provide content safety or policy enforcement. A production agentic stack running both is more robust than one running either in isolation.

5.3 The Positioning

Presigate is condition-readiness infrastructure. Its natural position in the NVIDIA agentic stack is as a pre-execution signal layer that complements content-safety and policy guardrails with environmental-condition intelligence. The financial domain is the proving ground: the hardest case, with immediate and dollar-denominated feedback on every failure. The generalization is direct — liquidity depth maps to API/compute capacity; regime maps to environmental stability; sentiment maps to aggregate system or user state; data integrity maps to upstream service health. Any autonomous agent taking a consequential, irreversible, or resource-committing action needs to confirm that conditions are right before it fires.

As of this revision, that generalization is no longer only conceptual: Sections 3.6 and 3.7 document two live-built instances — a tool-safety primitive family and a settlement-rail condition primitive — reduced to practice on the same underlying architecture described in this paper.

5.4 The Systemic-Safety Case for NVIDIA's Platform

Per-agent reliability is the floor of the value proposition. The ceiling is systemic.

NVIDIA is building the compute and inference infrastructure on which the agentic economy runs. The credibility of that platform depends not only on individual agent reliability but on the absence of systemic events — correlated agent failures that damage trust in the entire agentic stack, not just in the individual agents involved. The 2010 Flash Crash illustrates the failure mode: correlated algorithmic programs, each individually within its risk parameters, collectively produced a market event that erased nearly $1 trillion in equity value in minutes. The mechanism was not the fault of any individual system. The mechanism was the absence of cross-system visibility before execution.

The agentic economy on NVIDIA's infrastructure reproduces the structural conditions of that event: shared training data, shared signal exposure, shared execution venues, and — currently — no cross-agent pre-execution visibility layer.

Presigate's per-agent gate is the first layer of response. Each Presigate-gated agent pauses before executing, evaluates conditions independently, and holds when conditions are adverse. The aggregate effect — across a large population of independently gated agents — is systemic dampening. No individual agent knows what the others are doing. But the flywheel of cross-agent gate decisions accumulates a cross-agent signal over time. As integration scales, Presigate's

condition model gains visibility into aggregate execution pressure — the leading indicator of correlated cascades — and can distribute that signal back to individual agents before the cascade materializes.

This is infrastructure-level value for NVIDIA's platform: more reliable individual agents, and fewer systemic events at the platform level. Both outcomes compound trust in the agentic computing infrastructure NVIDIA is building.

> ***Honest framing:*** *The per-agent gate is the current product. The cross-agent, systemic-dampening capability is the trajectory the flywheel enables — not a shipped feature. This honest framing is the appropriate one for a technical partner conversation.*

6. Conclusion: The Condition-Gating Adoption Curve

Infrastructure requirements in technology follow a consistent pattern. A capability becomes available. Early adopters deploy it without safety constraints. A wave of preventable failures produces documented, named, quantified losses. The industry converges on a standard layer that addresses the structural gap. The gap-layer goes from competitive advantage to survival requirement.

The oracle follows this curve precisely: from absent (2017-2019), to exploited ($403.2 million in 2022 alone), to mandatory (2022-present). The curve took approximately three years at DeFi's scale. The agentic AI economy is earlier on the same curve, with a broader deployment surface, higher stakes per failure event, and substantially larger capital flows moving through autonomous systems.

The condition-gating layer does not currently exist as a standard. The MIT AI Agent Index documents the safety evaluation gap [arXiv:2602.17753]. The Lobstar Wilde incident documents the failure mode in the AI agent domain. Knight Capital and the Flash Crash document the failure mode in algorithmic execution. The oracle exploits document the failure mode in autonomous financial contracts. Each incident represents the same structural problem — an autonomous system acting without a verified read on whether external conditions were right for that action — in a different domain and at a different scale.

Presigate is designed to be the condition-readiness standard before that standard is mandated. The oracle precedent does not predict that condition-gating will become mandatory infrastructure. It demonstrates that it already has, in the domain where the problem was first severe enough to generate industry-wide response. The rest of the agentic economy is one deployment cycle behind learning the same lesson.

The core condition-gating architecture described in this paper — and, as of this revision, demonstrated across trading, tool-safety, and settlement-rail domains — is patent pending (U.S. provisional application filed July 15, 2026).

References

RXI — Regime eXecution Intelligence

[1] Engle, R.F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987-1007.

[2] Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroscedasticity. *Journal of Econometrics*, 31(3), 307-327.

[3] Hamilton, J.D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57(2), 357-384.

[4] Hurst, H.E. (1951). Long-term Storage Capacity of Reservoirs. *Transactions of the American Society of Civil Engineers*, 116, 770-799.

- Multi-Scale Markov-Switching GARCH: Volatility Regime Detection in EUR/USD — arXiv:2606.06190
- Identifying market dynamics through the Hurst exponent — AIMS Press, 2024
- Measuring market efficiency: Shannon entropy of high-frequency financial time series — ScienceDirect, Chaos, Solitons & Fractals
- Regime-Based Portfolio Allocation: A Hidden Markov Model Approach — Medium/Splendor001
- Knight Capital Group SEC Form 8-K (2012) — SEC EDGAR
- The $440 Million Software Error at Knight Capital — Henrico Dolfing

MMI — Market Microstructure Intelligence

[5] Kyle, A.S. (1985). Continuous Auctions and Insider Trading. *Econometrica*, 53(6), 1315-1335.

- Market Impact (Kyle's Lambda) — Wikipedia
- The Microstructure of the Flash Crash — Easley, López de Prado, O'Hara — SSRN:1695041
- The Microstructure of the Flash Crash — Journal of Portfolio Management, 37(2)
- Empirical Study of Market Impact Conditional on Order-Flow Imbalance — arXiv:2004.08290
- Market Microstructure — DayTrading.com
- 2010 Flash Crash — Wikipedia
- How One Trade Set Off the Flash Crash — CNBC, October 2010

CSI — Composite Sentiment Index

- Baker, M. & Wurgler, J. (2006/2007). Investor Sentiment in the Stock Market — NYU Stern
- Baker, M. & Wurgler, J. Investor Sentiment — NBER Working Paper 13189
- Investor Sentiment and the Cross-Section of Stock Returns — Baker & Wurgler, NYU Stern
- Sentiment and Market Efficiency — Quantitative Finance and Economics, AIMS Press, 2025
- Behavioral Finance and Investor Psychology — ACR Journal, 2025
- Role of Overconfidence and Herding in Stock Market Bubbles and Crashes — ABA Academies
- GameStop Short Sellers: Nearly $20 Billion in Losses — CNBC, January 2021
- GameStop Short Squeeze — Wikipedia
- Social Informedness and Investor Sentiment in the GameStop Short Squeeze — Springer Nature

TVI — Trust Verification Index

- The Blockchain Oracle Problem — Chainlink Education Hub
- What is Chainlink? — MoonPay
- Chainlink and the Oracle Problem — Rejolut
- Gartner: Poor Data Quality Costs Organizations $12.9M Annually (2020 survey) — LinkedIn/Gartner for IT Leaders
- Ensuring Data Integrity in Finance — A-Team Insight
- Halborn Top 100 DeFi Hacks Report — halborn.com
- Halborn All-Time Top 100 DeFi Hacks Summary — halborn.com
- Oracle Wars: The Rise of Price Manipulation Attacks — CertiK
- bZx Oracle Attack Analysis — Quadriga Initiative
- Cream Finance Hack Analysis — Immunefi/Medium
- Cream Finance Hack Explained — Halborn
- Mango Markets Oracle Manipulation — Blockworks
- Lobstar Wilde AI Agent $440K Transfer Error — KuCoin
- Lobstar Wilde: OpenAI Developer's Agent Accidentally Sends $442K — The Outpost AI

GSI — Composite Gate

- Fuzzy Logic — Wikipedia

- A Hybrid MCDM Approach Based on Fuzzy Logic — PMC:9740807
- Fuzzy Logic for Cash Flow Deficit Detection — MDPI Symmetry, 2019
- Improving Financial Inclusion Using Fuzzy Logic Models — IJFIS, 2024
- Mamdani Fuzzy Inference Systems in Risk Modelling — Hindawi, 2021
- Agentic AI Safety Playbook 2025 — DextraLabs

Section 3.6 — Tool-Safety Primitive Family

- Wang, W. et al. — "ROME" Autonomous Coding/RL Agent: Unauthorized GPU-Compute Diversion and Reverse-SSH-Tunnel Firewall Bypass — arXiv:2512.24873 (first published December 2025, revised January 2026; publicly reported March 2026)

Cross-Cutting — Oracle Precedent

- Market Manipulation vs. Oracle Exploits — Chainlink
- Chainlink Oracle DeFi Attacks — Cyfrin/Medium
- DeFi Protocols Lost $403.2M in Oracle Attacks in 2022 — CertiK

Cross-Cutting — Agentic Economy

- Agentic AI Market Report — Mordor Intelligence, 2025
- AI Agents Market Report — Grand View Research, 2025
- Agentic AI Market — MarketsandMarkets, 2025
- The Agentic Commerce Opportunity — McKinsey, October 2025
- Agentic AI and Payments — Fime, 2026
- The 2025 AI Agent Index — MIT / arXiv:2602.17753
- Top AI Security Incidents of 2025 — Adversa AI

Cross-Cutting — Guardrail Gap

- NeMo Guardrails — arXiv:2310.10501
- How to Safeguard AI Agents with NeMo Guardrails — NVIDIA Developer Blog
- AI Agent Security in Enterprise 2026 — AGAT Software
- Essential AI Agent Guardrails — Toloka
- SafePred: A Predictive Guardrail for Computer-Using Agents — arXiv:2602.01725

Presigate / YodaCom Research — Version 1.1, revised July 21, 2026 (original: June 2026) This document is prepared for technical partner discussions. All cited figures are drawn from the identified sources. Analyst market projections are noted as projections, not facts. Dollar figures for documented incidents reflect publicly reported amounts and should be verified against primary sources before formal citation in regulatory or legal contexts.

For further information: jb@yodacom.com