Nick Clark · Published June 29, 2026 · PDF

Cybersecurity Threat Forecasting: Simulating Adversary Trajectories and Predictive Network Reconfiguration as Non-Executing Speculation

Security operations centers face an unforgiving asymmetry: a defender that reasons about what an attacker might do next can be tricked into acting on that reasoning, turning a forecast into a self-inflicted outage or an automated misconfiguration. The defense agent needs to explore many speculative attack trajectories and many candidate network reconfigurations without ever letting an unproven hypothesis touch production. This application is built on the Forecasting Engine, disclosed in United States Patent Application 19/647,395, which contains speculative reasoning in a structurally separate domain and admits action only through a single governed promotion gateway.

What This Application Specifies

This application describes a defensive security agent whose threat forecasting is structured as a planning graph. As disclosed in United States Patent Application 19/647,395, a planning graph is a mutable, memory-referenced directed structure that represents one or more hypothetical future states. Its root node is the agent's current verified state and each branch is a distinct hypothetical trajectory: a sequence of speculative mutations, environmental transitions, or intent resolutions the agent is

evaluating as a possible future. The specification is explicit that a planning graph is not an execution plan, a schedule, or a commitment. It is a pre-execution construct that lives in a computational domain structurally distinct from verified execution memory.

Applied to a security operations context, each branch of the planning graph models one candidate adversary trajectory or one candidate defensive reconfiguration. An anomaly observed on the network seeds new branches; the forecasting engine projects each branch forward across speculative steps. Per the specification, every branch encodes a speculative mutation sequence, a projected outcome, an affective reinforcement tag, a trust slope projection, a policy compatibility flag, and a branch classification label. Nothing about constructing or scoring these branches changes the production network. The specification states that no mechanism exists by which a planning graph branch can directly modify verified execution memory without passing through the governance-validated promotion pathway.

The boundary is enforced, not advisory. The specification describes a containment layer that tags every element of a planning graph with an immutable speculative marker at the time of construction. That marker cannot be removed, modified, or overridden by any operation inside the planning graph domain. Only the promotion interface, upon successful governance validation, strips the marker and re-tags the content as verified before writing it to execution memory. There is a single gateway, and its governance requirements are described as not negotiable, waivable, or bypassable by the agent's state or operational urgency.

Why It Matters

The hard problem in autonomous defense is not generating hypotheses about an attacker. It is preventing the defender's own hypotheses from becoming actions before they are warranted. A SOC automation that reacts directly to a speculative attack chain can be weaponized: an adversary who understands the trigger can feed the system

anomalies engineered to provoke a damaging automated response, such as isolating a critical segment, revoking legitimate credentials, or rerouting traffic into a chokepoint. The forecast becomes the attack surface.

The specification addresses this by making speculation structurally inert. Because planning graph branches carry an immutable speculative marker and live in a separate domain, the agent can explore an adversary's most aggressive projected trajectory, and the most aggressive candidate countermeasure, without either one acquiring the status of a fact. The specification notes that the separation lets the agent maintain multiple contradictory hypothetical futures simultaneously without internal inconsistency: one branch projecting a breach succeeds, another projecting it is contained, and neither perturbs verified state because neither has been promoted. For threat hunting this is exactly the property required to game out an intrusion to its conclusion without committing the network to any of the imagined outcomes.

How It Composes With the Domain

Anomaly-driven branch generation maps onto the specification's instantiation logic, which reads the agent's intent field and current verified state to construct branches. In the security setting, a detected anomaly defines the objective the planning graph explores: what trajectories could produce this signal, and what reconfigurations could blunt them. The number of speculative alternatives explored at each step is the branching factor, which the specification ties to the agent's modulated state; the depth is how many steps into the future the engine projects before terminating exploration.

The decisive filter for any candidate defensive action is slope-constrained speculative simulation. The specification describes the trust slope trajectory as a hard architectural boundary, not a soft preference, that prevents promotion of any branch whose execution would produce a trust slope discontinuity. For each branch, the slope validation module computes a hypothetical Derived Anchor Hash by applying the branch's speculative mutation sequence to a sandboxed copy of the agent's lineage, then

checks continuity against the current trajectory using the same validation the governance infrastructure applies to committed mutations. A candidate reconfiguration whose provenance chain would break is slope-ineligible and cannot advance. Critically, the specification states this constraint operates prospectively: it filters branches before they reach the promotion interface, so the governance pipeline never receives a candidate that would fail validation.

Branches are then classified. The specification's taxonomy gives four categories. An eligible branch has passed slope validation, satisfied policy compatibility, and received positive or neutral reinforcement; it is a viable promotion candidate, ranked by a composite score over projected outcome quality, trust slope continuation, integrity impact, reinforcement strength, and intent alignment. An introspective branch passed slope and policy but is reinforced negatively and is retained for self-examination rather than promotion, which lets a defense agent keep a structurally viable but undesirable response on the table for reasoning without making it actionable. A delegable branch is better handed to a specialized child agent. A pruned branch failed validation or was superseded and is scheduled for removal. The specification stresses that classification is not permanent: branches are re-evaluated each cycle as conditions change, so a foreclosed countermeasure can become eligible if the situation shifts.

Predictive network reconfiguration is grounded directly in Section 6.10. The specification describes a predictive network planning subsystem that uses temporal health forecasts and capability pressure trajectories to simulate the impact of infrastructure changes before those changes are enacted. It accepts proposed changes as input, such as taking a substrate offline or redeploying a model, and simulates the effect on capability pressure and the temporal health forecast, producing quantitative projections. The same section describes automated reconfiguration that the system may enact only when authorized by governance policy, including rerouting work to substrates whose envelopes are projected to open sooner. In the security framing, this is the difference between simulating a segment isolation and performing one: the simulation is contained speculation; the enactment requires governed dispatch.

Dispatch itself is confidence-gated. The specification's confidence governor evaluates execution readiness; where confidence is insufficient it transitions the agent into a non-executing cognitive mode in which speculative reasoning, planning, and inquiry generation continue without committing state changes. For a SOC, that means a defense agent under genuine uncertainty keeps forecasting attacker moves and keeps building candidate responses while declining to act, and can raise an inquiry instead of executing.

What This Enables

A defense agent built on this disclosure can fully simulate an intrusion, branching on each observed anomaly into the trajectories that could explain it and the reconfigurations that could counter it, with a structural guarantee that none of that simulation alters the network. It can hold contradictory hypotheses about an attacker at once, score candidate countermeasures against trust slope continuity and projected integrity impact, and surface the leading eligible response without enacting it. When it does act, the action passes through one auditable promotion gateway whose decision is recorded in the agent's lineage, giving incident responders a provenance trail for every automated change. Under uncertainty it degrades into continued forecasting and inquiry rather than into reckless action or paralysis.

Boundary Conditions

The specification is a disclosure of cognitive architecture, not a detection product. It does not specify intrusion-detection signatures, threat-intelligence feeds, packet inspection, or detection accuracy rates, and none should be inferred here. The quality of the forecasts depends entirely on the quality of the anomaly signals and verified state fed to the instantiation logic; the architecture governs how speculation is contained and dispatched, not how threats are sensed. Slope-constrained simulation foreclosing a branch means it cannot be promoted to action, not that the underlying adversary

behavior is impossible; foreclosed branches may be retained only for introspection. Automated reconfiguration is gated on governance authorization, so an operator's policy floor, not the agent's urgency, sets the limit of autonomous action. The containment guarantees hold under the architectural assumptions of the specification, including the integrity of the immutable speculative markers; the specification itself discusses containment collapse as a failure mode to be detected and recovered from rather than assumed away.

Disclosure Scope

The cognitive architecture described here, including planning graphs, the containment boundary with immutable speculative markers, slope-constrained speculative simulation, branch classification, predictive network planning and reconfiguration, and confidence-gated dispatch into a non-executing cognitive mode, is disclosed in United States Patent Application 19/647,395. The security operations framing in this article, including SOC threat hunting, adversary-trajectory modeling, and predictive defense, is external domain context provided to illustrate an enabling implementation; it is not part of the disclosure and adds no new technical mechanism. Any references to operational security practices, regulatory expectations, or network-defense workflows are descriptive context only. Every claim in this article about what the invention does traces to the specification of United States Patent Application 19/647,395.

Forecasting Engine (/forecasting-engine) All 40 steps → (/inventive-steps)

Plan before you act. Contain speculation. Promote only what passes.

Chapter 4 (/patents/19-647395/chapters/forecasting)

PRIMARY TECHNICAL DISCLOSURE

- Forecasting and Executive Graphs in Autonomous Cognitive Systems (/articles/forecasting-and-executive-graphs-in-autonomous-cognitive-systems)

SECONDARY TECHNICAL

- Planning Graphs as First-Class Cognitive Structures (/articles/forecasting-engine/planning-graphs)
- Containment Layer and Delusion Boundary (/articles/forecasting-engine/containment-boundary)
- Branch Classification System (/articles/forecasting-engine/branch-classification)
- Personality Field as Structural Modifier (/articles/forecasting-engine/personality-modifier)
- Executive Engine Multi-Agent Graph Aggregation (/articles/forecasting-engine/executive-aggregation)
- Branch Dormancy and Deferred Promotion (/articles/forecasting-engine/branch-dormancy)
- Proactive Speculative Maintenance (Dream State) (/articles/forecasting-engine/dream-state)
- Planning Graph Archival for Cognitive Forensics (/articles/forecasting-engine/cognitive-forensics)
- Cross-Agent Planning Graph Visibility (/articles/forecasting-engine/cross-agent-visibility)
- Slope-Constrained Speculative Simulation (/articles/forecasting-engine/slope-constrained)
- Structural Separation From Verified Memory (/articles/forecasting-engine/memory-separation)
- Forecasting Engine Architecture (/articles/forecasting-engine/architecture)
- Forecasting Execution Cycle (/articles/forecasting-engine/execution-cycle)
- Emotional Modulation of Planning (/articles/forecasting-engine/emotional-modulation)
- Executive Graph Conflict Resolution (/articles/forecasting-engine/conflict-resolution)
- Planning Graph Delegation and Forking (/articles/forecasting-engine/delegation-forking)
- Temporal Anchoring and Lifecycle Management (/articles/forecasting-engine/temporal-anchoring)
- Forecasting as Coordination Primitive (/articles/forecasting-engine/coordination-primitive)
- Forecasting-Modulated Discovery Traversal (/articles/forecasting-engine/discovery-shaping)
- Forecasting as Confidence Input (/articles/forecasting-engine/confidence-input)
- Integrity-Constrained Forecasting (/articles/forecasting-engine/integrity-constrained)
- Forecasting for Training Curriculum (/articles/forecasting-engine/training-curriculum)
- Biological Signal to Forecasting Coupling (/articles/forecasting-engine/biological-forecasting)
- Substrate-Agnostic Forecasting Deployment (/articles/forecasting-engine/substrate-deployment)
- Uncertainty-Driven Solicitation in the Forecasting Engine (/articles/forecasting-engine/uncertainty-driven-solicitation)

- Cascade Forecasting in the Planning Graph (/articles/forecasting-engine/cascade-forecasting)
- Fleet Behavior Extrapolation (/articles/forecasting-engine/fleet-behavior-extrapolation)

APPLICATIONS · GENERAL

- **Cybersecurity Threat Forecasting: Simulating Adversary Trajectories and Predictive Network Reconfiguration as Non-Executing Speculation (/articles/forecasting-engine/cybersecurity-threat-forecasting)**
- Surgical Robot Planning AI: Safe Speculative Planning That Never Reaches the Patient (/articles/forecasting-engine/surgical-planning)
- AI Tactical Planning That Explores Adversary Options Without Committing Forces (/articles/forecasting-engine/defense-tactical-planning)
- AI Logistics Planning That Keeps Contingencies Ready: Governed Planning Graphs for Supply Chain Operations (/articles/forecasting-engine/logistics-planning)
- AI Disaster Response Planning: Multi-Scenario Resource Allocation Under Uncertainty (/articles/forecasting-engine/disaster-response-planning)
- Forecasting Engine for Financial Portfolio Planning (/articles/forecasting-engine/financial-portfolio-planning)
- AI Schedule Contingency Management for Construction Project Delay Recovery (/articles/forecasting-engine/construction-project-planning)
- Epidemic Response Planning AI: Multi-Scenario Outbreak Forecasting With an Auditable Decision Record (/articles/forecasting-engine/epidemic-response-planning)
- AI Space Mission Planning: Trajectory Branching and Abort Forecasting Under Light-Time Delay (/articles/forecasting-engine/space-mission-planning)
- Fleet-Scale Active Perception for Autonomous Vehicle Compliance (/articles/forecasting-engine/active-perception-fleet)
- Smart-Grid Load Forecasting With Contained Speculative Planning Graphs (/articles/forecasting-engine/smart-grid-forecasting)

APPLICATIONS · SPECIFIC

- Intuitive Surgical da Vinci vs Governed Forecasting: Trajectories, Not Consequences (/articles/forecasting-engine/intuitive-surgical)
- Anduril Lattice vs Governed Mission Planning: Speculative Containment (/articles/forecasting-engine/anduril)
- Boston Dynamics vs Governed Mission Planning: Motion Is Not Cognition (/articles/forecasting-engine/boston-dynamics)
- Shield AI Hivemind vs Governed Speculative Planning: The Forecasting Engine Axis (/articles/forecasting-engine/shield-ai)

- MuJoCo vs Governed Robot Planning: Contained Speculation Above the Physics Simulator (/articles/forecasting-engine/mujoco)
- NVIDIA Isaac Sim vs Governed Agent Planning: The Forecasting Engine Gap (/articles/forecasting-engine/nvidia-isaac)
- Unity ML-Agents vs Governed Agent Planning at Runtime (/articles/forecasting-engine/unity-ml)
- Gazebo Alternative for Governed Robot Planning: Simulate the World, Contain the Cognition (/articles/forecasting-engine/gazebo)
- Drake vs Governed Robot Planning: Beyond Trajectory Optimization (/articles/forecasting-engine/drake)
- robosuite alternative for governed manipulation planning (/articles/forecasting-engine/robosuite)
- Mobileye REM vs Governed Speculative Planning: Where a Contained Forecasting Layer Sits Above the Roadbook (/articles/forecasting-engine/mobileye-rem)
- Tomorrow.io vs Governed Agent Forecasting: Two Meanings of Forecast (/articles/forecasting-engine/tomorrow-io)
- Skydio vs. a self-forecasting AI agent: trajectory forecasting in flight versus in cognition (/articles/forecasting-engine/skydio)

Forecasting Engine overview → (/forecasting-engine)