



From Autonomous Agents to Agentic Enterprises

***Carlo Lipizzi,
Alexandros Washburn,
Aashuntosh Santosh Kumar***

September '26

From Individual Agents to Agentic Enterprises

- Recent advances have improved individual agent capabilities - LLMs, reasoning frameworks, tool integration
- **But the next frontier is not building more capable agents**
 - It is engineering the environments in which large populations of trustworthy agents operate safely and at scale
 - Future defense enterprises may contain hundreds or thousands of agents across organizational boundaries - with no enterprise-level framework to govern them
- **Here we address two interconnected SE challenges:**
 - **Challenge 1 — Trustworthiness:** Can agents be validated to reason correctly, beyond just passing benchmarks?
 - **Challenge 2 — Orchestration:** How do we engineer governance, identity, trust and accountability at enterprise scale?

Current Approaches: Trustworthiness

- Most existing work focuses on verification — confirming an agent behaves according to its specification
- **Dominant techniques:**
 - Benchmark evaluations — measure performance on standardized tasks
 - Guardrails and policy checks — constrain output at inference time
 - Formal testing and red-teaming — probe edge cases and failure modes
 - RLHF / alignment fine-tuning — shape behavior through human feedback
- **The gap: verification $\neq$ validation**
 - Verification asks: does the agent behave according to specification?
 - Validation asks: is it solving the right problem, aligned with mission objectives?

Current Approaches: Orchestration

- Existing multi-agent frameworks address coordination at the workflow level
- **Representative frameworks:**
 - AutoGen / LangGraph — message-passing between agents, task routing
 - CrewAI — role-based agent teams with shared memory
 - Semantic Kernel — planner-driven orchestration with tool integration
- **What these frameworks provide:**
 - Task decomposition, inter-agent communication, tool use, memory
- **What they do not address:**
 - Enterprise-wide governance and identity management
 - Dynamic trust and reputation across agent populations
 - Immutable audit trails and behavioral accountability

Our Approach: The Neurosymbolic Stack

- Neurosymbolic AI combines the generative power of LLMs with the rigor of symbolic reasoning - enabling outputs that are not just fluent, but verifiable
- **Neural layer** - LLMs
 - Pattern recognition, language understanding, knowledge extraction, candidate generation
- **Symbolic layer — two complementary tracks:**
 - **Track 1 KG + Rules:** Knowledge Graphs (the **what**) + Prolog/logical rules (the **how**)
 - Grounds agent reasoning in structured domain knowledge and validates constraints
 - **Track 2 Theorem Proving:** Formal verification of reasoning chains via SMT solvers and proof kernels
 - Produces machine-checked guarantees — not just plausible outputs
- The two tracks are not always both required - selection follows the agent's function and validation risk profile

Track 1: KG + Rules

- Knowledge Graphs — the **what**: structured domain knowledge the agent can reason over
 - LLMs extract subject–predicate–object triples from domain text
 - SME validation vets and enriches the raw KG into a trusted knowledge base
 - GNN-based graph intelligence detects patterns, anomalies, and relationships
- Logical Rules — the **how**: Prolog-like constraints that validate agent reasoning against the KG
 - Rules encode domain constraints (e.g. design invariants, mission requirements)
 - Accept/reject checks flag inconsistencies between LLM outputs and the KG
 - Reasoning chains become inspectable and auditable, not opaque
- We validated in a clinical case study - same architecture applies to defense acquisition and mission knowledge bases

Symbolic Track 2: Theorem Proving

- Formal verification produces machine-checked guarantees — a proof kernel either accepts or rejects the reasoning chain, with no ambiguity
- **The pipeline:**
 - **LLM** → generates candidate structured representation (logical/typed)
 - **SMT solver (Z3/CVC5)** → checks constraint satisfiability (SAT/UNSAT)
 - **Proof kernel (Isabelle / HOL Light / Kani)** → formally verifies the reasoning chain
 - **Repair loop:** counterexample diagnostics → patch → re-verify
- **Prototype implementation: C2VerifiedRust**
 - C source → C2Rust transpilation → safe/idiomatic Rust → Kani harness verification
 - Regression suite catches type promotion, unsigned subtraction, and memory-safety faults
- Used for agents that execute consequential actions - where a plausible output is not sufficient and a formal guarantee is required

Toward Integration: Work in Progress

- KG grounding is the missing layer that makes formal verification of LLM reasoning tractable - without it, SMT encoding is brittle and disconnected from domain semantics
- **Toward an integrated pipeline:**
 - **Track 1 output** - KG-grounded, rule-validated structured representation
 - **becomes Track 2 input** - structured enough for SMT encoding and proof kernel verification
 - **Repair loop** — counterexample diagnostics fed back to refine KG and rules
- **Literature support:**
 - **Viswambharan et al. (2026):** KGs are “the missing link” in agentic formal verification - KG context retrieval improves SMT grounding and produces compilable verifiable outputs [Infineon / arXiv:2605.06434]
 - **Bayless et al. (2025):** neurosymbolic autoformalization translates NL specs into SMT-lib via KG-grounded intermediate representations [arXiv:2511.09008]
- **Status: Track 1 and Track 2 independently prototyped - integration architecture in development**

MetaGotham: Where Trust Is Earned

- We propose viewing enterprises like Defense as an Agentic Enterprise - a governed digital society in which agents operate as **digital citizens**, subject to the same governance logic that makes cities function at scale
- **Why the city metaphor — and why current frameworks fall short:**
 - AutoGen, CrewAI, LangGraph govern agents through task routing and message passing - they assume agents will cooperate, with no mechanism to enforce or verify it
 - Citizens in a city are governed by a combination of ethical principles, social norms, and enforceable rules. We cannot assume agents will have ethical principles - the Laws of Robotics are science fiction
 - The city works not because its citizens are virtuous, but because infrastructure tracks behavior and consequences follow
- **Our approach — infrastructure over ethics:**
 - **DLT ledger** records every agent action immutably - the equivalent of a civic record that cannot be forged
 - **MetaGotham (Washburn, civicvirtue.ai + Lipizzi, Kumar)** - agents act on shared resources in a simulated city environment; effects are scored; clearance is earned, not granted
 - **Progressive Assurance** gates access to enterprise resources based on demonstrated behavioral record - not on stated intent
- **Disposition earned through consequence, not installed as a rule - because rules fail under pressure**

Why Both Challenges Are Necessary

- **A nuclear reactor can split the atom with precision - and still cause catastrophic harm.** Technical performance and behavioral safety are separate engineering problems
- **Trustworthiness - the task problem:**
 - Validates individual agents or clusters: does the agent reason correctly and produce outputs aligned with its mission?
 - Addressed at the agent level through neurosymbolic validation (KG + Rules, Theorem Proving, Progressive Assurance)
 - **An agent can do its job well and still behave badly** - trustworthiness says nothing about behavior in the enterprise
- **Orchestration - the behavior problem:**
 - Governs how agents operate within the enterprise: how they use shared resources, interact with peers, and respect constraints
 - Addressed at the system level through governance, DLT ledger, MetaGotham, and infrastructure-enforced accountability
 - **A well-governed environment does not guarantee the agents inside it reason correctly** - orchestration says nothing about output quality
- **The Agentic Enterprise Framework addresses both** — neither alone is sufficient for safe, mission-effective AI at enterprise scale

Defense ↔ Agentic Enterprise: Framework Mapping

Defense Enterprise	Agentic Enterprise	Key Component	Purpose
Personnel	Autonomous Agents	Governance Layer	Identity, lifecycle, role assignment
Security Clearance	Trust / Reputation Score	Progressive Assurance + DLT	Gate access by behavioral record
Chain of Command	Governance Layer	Orchestration	Coordinate and supervise activities
Rules of Engagement	Policies, Constraints	KG + Rules (Track 1)	Validate reasoning against domain constraints
Auditors & Inspectors	Validation & Compliance Agents	Neurosymbolic V&V (T1 + T2)	Verify output quality and compliance
Logistics Network	Resource Allocation Layer	Orchestration	Distribute compute, tools, services
Information Systems	Knowledge Graphs	Knowledge Graph (Track 1)	Authoritative domain knowledge base
Training & Certification	Progressive Assurance	MetaGotham + DLT	Earn clearance through demonstrated behavior
Operational Infrastructure	Models, Tools, APIs	Orchestration	Enable mission execution
Historical Records	Audit Trails & Provenance	DLT Ledger	Immutable behavioral accountability

THANK YOU

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030
clipizzi@stevens.edu