Reconsidering Architectures for Information Fusion Systems in CopForge: AI Flows and AI Agents

CINN - TI/IA

January 12, 2026

Abstract

This document examines the concept of agency in Large Language Model (LLM)-based systems and argues that agency is best understood as a spectrum rather than a binary classification. Within the CopForge information ingestion and fusion domain, we analyze two architectural options: (i) an *AI Flow*, implemented as a fixed pipeline or DAG with conditional routing, and (ii) an *AI Agent*, implemented as an LLM-driven reasoning loop with autonomous tool selection. We compare these approaches along dimensions such as control flow, tool autonomy, latency, cost, predictability, and fusion quality in open-ended correlation scenarios. Rather than positioning one architecture as universally superior, we show that AI Flows favor throughput and determinism, while AI Agents favor contextual reasoning and flexibility, making each suitable for different operational contexts.

1 Introduction: The Spectrum of Agency

The term "agent" has become ubiquitous in the field of artificial intelligence, yet its definition varies substantially across different frameworks and research communities. Rather than constituting a binary property, agency can be more accurately conceptualized as a *spectrum* that ranges from purely deterministic functions to fully autonomous systems capable of self-directed goal pursuit. Understanding where a given system falls on this spectrum has significant implications for architectural decisions, capability expectations, and the selection of appropriate communication protocols.

Anthropic employs a relatively strict definition of agency. According to this perspective, a true agent must exhibit: (i) significant autonomy in decision-making, (ii) the capacity for planning and goal-directed behavior, (iii) the ability to select and use tools based on self-determined necessity, and (iv) persistence of objectives across multiple interactions. Under this definition, many systems commonly labeled as "agents" would more accurately be classified as tool-augmented language models or orchestrated pipelines. However, alternative perspectives exist in the literature and industry practice. LangChain defines an agent as any system that uses an LLM to decide which actions to take. CrewAI conceptualizes agents as specialized roles equipped with objectives and tool access. The classical AI definition from Russell and Norvig [1] characterizes an agent as any entity that perceives its environment and acts upon it. In the reinforcement learning paradigm, an agent is defined as an entity that maximizes cumulative reward through sequential action selection.

These divergent definitions reflect genuine differences in design philosophy and intended use cases. The key insight is that agency admits of degrees, and the appropriate level of agency depends on the requirements, constraints, and trade-offs of the application.

2 The Agency Spectrum: A Taxonomy

To operationalize the concept of agency as a spectrum, we propose a taxonomy based on the degree of autonomous decision-making exhibited by a system. Agency varies along at least four

orthogonal dimensions:

1. **Control of Execution Flow**
   - Externally controlled: execution order is fixed (pipelines, DAGs).
   - Conditionally routed: limited branching determined by predefined rules.
   - Model-controlled: the LLM decides which step to execute next.
2. **Knowledge Source**
   - Parametric only: the model relies solely on its internal weights.
   - Augmented: retrieval, databases, or external state supplement the model.
3. **Tool Autonomy**
   - No tools: pure text generation.
   - Callable tools: tools invoked only when explicitly instructed.
   - Selectable tools: the model decides which tool to use and when.
4. **Temporal Coherence**
   - Stateless: each invocation is independent.
   - Session-scoped: short-term memory within a request.
   - Persistent: objectives and context span multiple interactions.

A system becomes more agentic not when it merely uses an LLM, but when control flow, tool choice, and temporal structure migrate from code into model reasoning.

3 Option A: AI Flow as a Deterministic Semantic Transformer

The TIFDA project [7] (Tactical Information Fusion and Dissemination Agent) and the early CopForge [8] architecture adopt an AI Flow architecture for information ingestion and fusion with the Common Operational Picture (COP). In this approach, sensor messages traverse a fixed sequence of processing stages:

1. **Firewall**: Security validation (injection detection, coordinate validation)
2. **Parser**: Format-specific parsing (ASTERIX, drone telemetry, radio intercepts)
3. **Multimodal**: Processing of associated files (audio transcription, image analysis)
4. **LLM Extract**: Entity extraction and enrichment using a language model
5. **Validate**: Output validation against the EntityCOP schema
6. **COP Update**: Insertion into the Common Operational Picture

Implemented as a directed acyclic graph (DAG), this architecture (Figure 1) places the LLM inside a predefined execution structure. While the LLM performs semantic interpretation, it does not control execution order or tool selection. This pattern is best characterized as a *deterministic semantic transformer*: a system in which semantic understanding is delegated to the model, but decision-making and control flow remain externally defined.

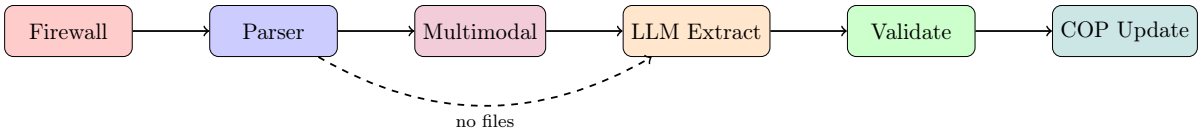

Figure 1: AI Flow architecture: a deterministic semantic transformer implemented as a fixed DAG

Such systems favor predictability, auditability, and throughput. They occupy a lower-to-middle position on the agency spectrum by design, which is advantageous in scenarios dominated by high-volume, low-complexity data streams.

4 Option B: AI Agent as a Reasoning Loop

As an alternative, CopForge can be implemented as an AI Agent based on an LLM reasoning loop. In this architecture (Figure 2), the model iteratively reasons about the task, selects tools, observes results, and decides whether to continue or terminate.

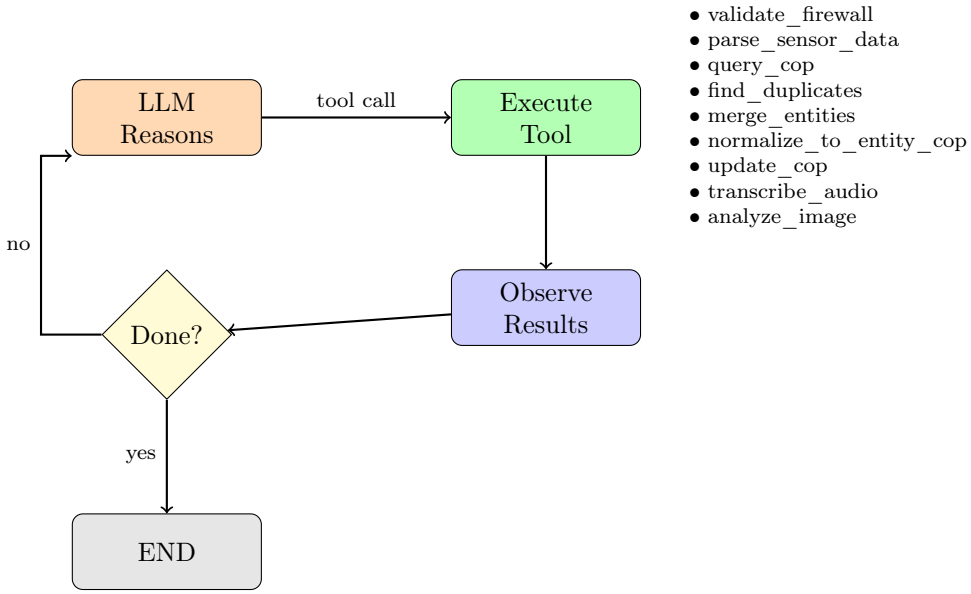

Figure 2: AI Agent architecture: LLM-driven reasoning loop with autonomous tool selection

This architecture enables contextual decision-making, proactive duplicate detection, adaptive processing strategies, and open-ended multi-sensor correlation. These capabilities come at the cost of increased latency, computational overhead, and reduced predictability.

5 Comparison and Trade-offs

Table 1 summarizes the key differences between the pipeline and agent architectures.

Characteristic	AI Flow	AI Agent
Control flow	Fixed DAG	Model-controlled
Tool selection	Predetermined	Autonomous
COP context queries	Not used	Proactive
Duplicate detection	Post-hoc only	Proactive
LLM calls per message	1	3–10
Latency	2–3 seconds	10–30 seconds
Cost per message	Low	Higher
Predictability	High	Medium
Auditability	Straightforward	Requires structured logging
	Easy (fixed flow)	Requires logging
Multi-sensor correlation	Limited	High

Table 1: Comparison of AI Flow and AI Agent architectures

6 Conclusion

Agency in LLM-based systems is best understood as a spectrum rather than a binary classification. Within CopForge, both AI Flows and AI Agents represent valid and complementary architectural choices. AI Flows, implemented as deterministic semantic transformers, favor efficiency, predictability, and high throughput. AI Agents extend the design space toward contextual reasoning and adaptive fusion at the cost of additional overhead.

The appropriate choice depends on workload characteristics, acceptable latency and cost envelopes, and the degree to which fusion logic can be specified at design time. In practice, hybrid architectures that combine AI Flows for routine processing with AI Agents for complex or ambiguous cases can provide a balanced solution.

References

[1] Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.

[2] LangChain. (2024). *Introduction to Agents*. https://python.langchain.com/docs/concepts/agents/

[3] Anthropic. (2024). *Building effective agents*. https://www.anthropic.com/research/building-effective-agents

[4] CrewAI. (2024). *Documentation*. https://docs.crewai.com/

[5] Anthropic. (2024). *Model Context Protocol Specification*. https://modelcontextprotocol.io/

[6] Google. (2025). *Agent-to-Agent Protocol*. https://github.com/google/A2A

[7] Martínez-Agulló, P. (2025). *TIFDA: GenAI-Enabled Tactical Information Fusion and Dissemination Agent*. https://github.com/MartinezAgullo/genai-tifda

[8] Martínez-Agulló, P. (2025). *COP Forge: Information Fusion System for Common Operational Pictures*. https://github.com/MartinezAgullo/copforge