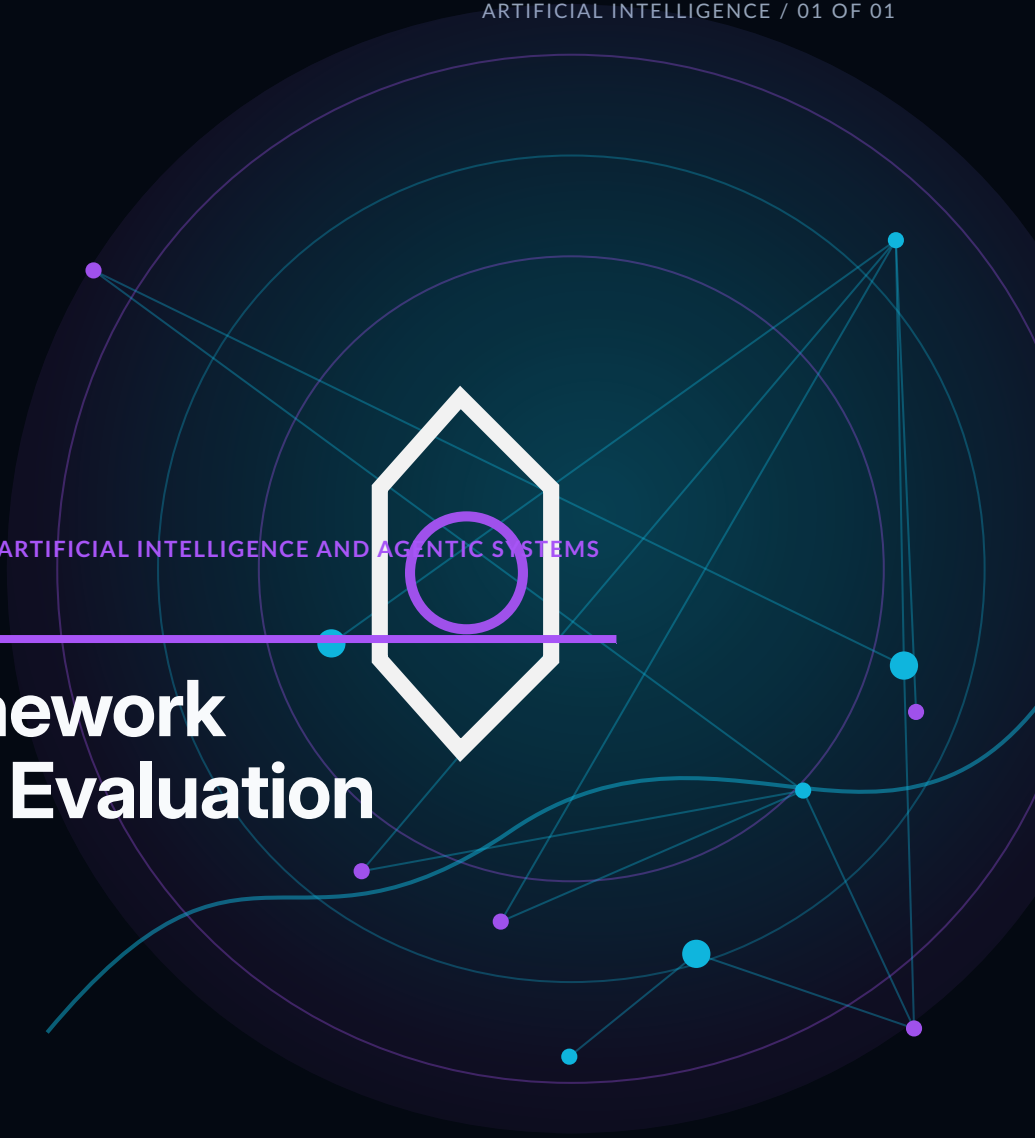


ENGINEERING STANDARDS LIBRARY / ARTIFICIAL INTELLIGENCE AND AGENTIC SYSTEMS

Agentic Framework Comparative Evaluation Companion

A profile-based decision companion for agent frameworks, durable runtimes, multi-agent systems, and managed platforms.

PERMANENT DOCUMENT ID

MYTHOS-CMP-COMP-AI-0001

PUBLICATION

Candidate Comparative Evaluation Companion
Version 1.0.0-candidate

ASSURANCE PROFILE

Standard / Advanced
High Assurance

PERMANENT PUBLICATION IDENTITY

Agentic Framework Comparative Evaluation Companion

DOCUMENT ID
MYTHOS-CMP-COMP-AI-0001

VERSION
1.0.0-candidate

STATUS
Candidate Comparative Evaluation Companion

PUBLISHER
Mythos Systems

PUBLICATION DATE
2026-07-24

SCHEDULED REVIEW
2027-01-24

CLASSIFICATION
Public

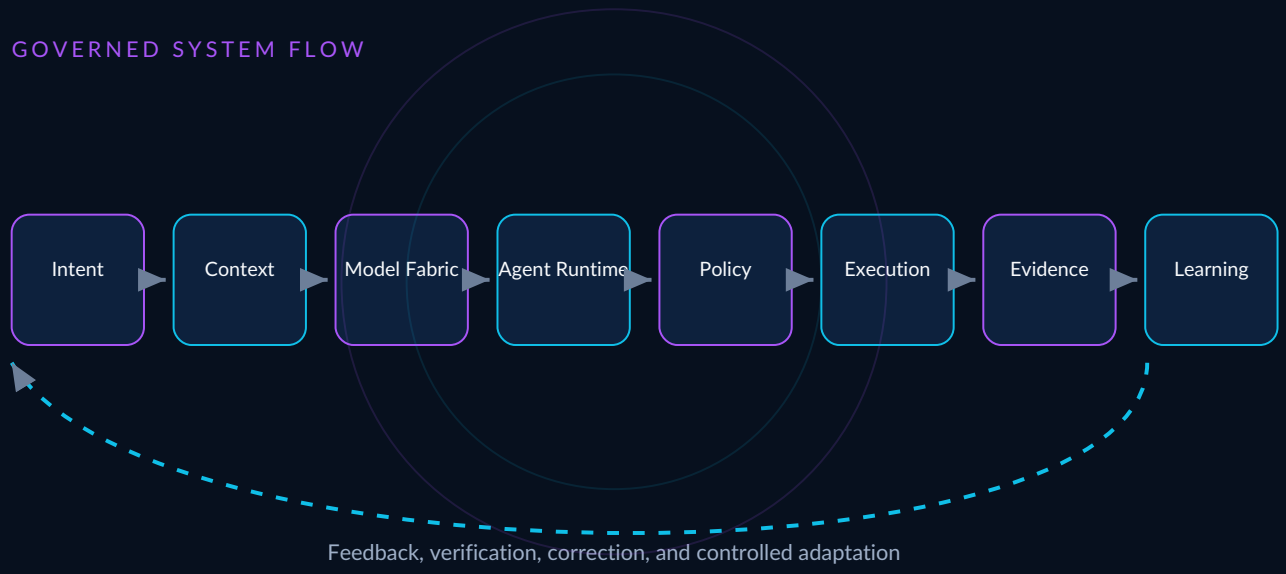
0	6	4	1
ATOMIC CRITERIA	ASSURANCE VIEWS	IMPLEMENTATION PROFILES	PERMANENT IDENTITY

Publication doctrine. This publication establishes durable guidance or requirements for its defined scope. Interpretation must preserve the permanent document identity, required evidence, controlled exceptions, source currency, and formal revision history.

COGNITIVE AND CONTROL TOPOLOGY

Artificial Intelligence and Agentic Systems

A trustworthy system does not reduce to a model, database, identity provider, or curriculum. It is a governed flow of intent, state, authority, execution, observation, evidence, correction, and controlled learning.



AUTHORITY

Reasoning may propose. Policy authorizes. Deterministic mechanisms execute. Evidence records the outcome.

EVIDENCE

Every consequential claim must resolve to a source, measurement, trace, test, signed artifact, or documented uncertainty.

STATE

Memory, identity, model behavior, and data lineage require explicit lifecycle, ownership, retention, and correction semantics.

LEARNING

Adaptation is gated by evaluation, provenance, regression protection, human oversight, and reversible release controls.

INTERPRETATION AND ASSURANCE

How to apply this publication

Normative intent

Requirements define outcomes and constraints. Implementations may vary, but evidence must demonstrate equivalent control.

Evidence over assertion

Vendor documentation and design intent are not equivalent to reproduced behavior. Record evidence grade and uncertainty.

Risk-proportional depth

Higher consequence, autonomy, sensitivity, and irreversibility require stronger isolation, review, verification, and recovery.

Controlled exception

Exceptions require an owner, rationale, compensating control, residual-risk acceptance, expiration, and reevaluation trigger.

Normalized assessment scale

0	1	2	3	4	5
ABSENT	AD HOC	PARTIAL	OPERATIONAL	ADVANCED	LEADING

A numeric score must never hide a failed mandatory gate, missing evidence, untested workload, or unacceptable residual risk.

INTERACTIVE NAVIGATION

Contents

Decision doctrine

This evaluation companion applies a profile-based benchmark. No framework is a universal winner. Mandatory gates, evidence confidence, durability, security, interoperability, operating model, and exit cost must remain separate from a weighted capability score.

Evaluation model

Dimension	Decision question
Mandatory gates	Does the platform meet non-negotiable security, identity, evidence, licensing, and deployment constraints?
Capability	How completely does it implement the required runtime and developer capabilities?
Evidence confidence	Are claims documented, reproduced, observed at scale, or independently verified?
Workload fit	Which task, risk, latency, data, and operating profiles does it support?
Transition risk	What is the cost of migration, state conversion, retraining, and provider exit?

Current candidate evaluation source

The following source evaluation is retained as research lineage and must be revalidated against current product versions and controlled proofs of concept before procurement or architecture decisions.

CMP-004: Agentic Frameworks (LangGraph vs. Bedrock AgentCore vs. Itential)

Document ID: CMP-004
Version: 1.0.0 (Candidate Library Edition)
Status: Candidate Comparative Evaluation
Classification: Public
Organization: Mythos Systems (Mythos Systems)
Owner: Aaron Stovall
Effective Date: 2026-07-16
Applies To: All organizations evaluating, selecting, or operating AI agentic frameworks, managed agent runtimes, and intelligent infrastructure orchestration platforms
Companion Standards: MYTHOS-AI-0001 (Agentic Framework Benchmark), MYTHOS-PLT-0004 (Managed Platform & Agentic Service Evaluation), MYTHOS-DAT-0004 (Event Sourcing & Durable State), RA-001 (Governed Multi-Agent Runtime)

0. Executive Summary

The architecture of autonomous orchestration has fractured into three distinct philosophies. Organizations are forced to choose between building complex cognitive loops from scratch (LangGraph), surrendering their core IP to a proprietary cloud managed runtime (AWS Bedrock AgentCore), or attempting to drag legacy network automation into the AI era (Itential).

Choosing the wrong framework is a multi-year architectural commitment. If the framework lacks durability, agents die on process crashes. If the framework couples tool execution to the vendor's cloud IAM, the organization surrenders its zero-trust boundary. If the framework relies on visual drag-and-drop builders, the logic becomes unmanageable "spaghetti" that cannot be version-controlled or formally verified.

This evaluation analyzes LangGraph, Bedrock AgentCore, and Itential against the rigorous demands of the Mythos Agentic Framework Standard (MYTHOS-AI-0001), which mandates strict authority separation, durable execution, and zero vendor lock-in.

1. The Competitors

1.1 LangGraph (by LangChain)

LangGraph represents the **"Build-It-Yourself" (Code-First)** paradigm. It is an open-source library that allows developers to define agent workflows as stateful, cyclical graphs. It provides the primitives (nodes, edges, state) but leaves the orchestration, crash recovery, and security boundaries entirely to the developer.

1.2 AWS Bedrock AgentCore (Managed Cloud Runtime)

Bedrock AgentCore represents the **"Managed Cloud Black Box"** paradigm. It is a fully managed AWS service that handles the heavy lifting of agent orchestration, multi-tenant scaling, and basic tool execution. It promises instant production-readiness but deeply couples the agent's cognitive loop and authority boundary to AWS infrastructure and IAM.

1.3 Itential (Enterprise Automation Platform)

Itential represents the **"Low-Code Infrastructure Orchestration"** paradigm. Itential is not a native LLM cognitive loop; it is a legacy network/IT automation platform that has bolted on AI capabilities. It excels at orchestrating complex workflows across thousands of network devices and APIs using visual builders, but it is fundamentally a deterministic workflow engine, not a probabilistic agent runtime.

2. Feature and Architecture Comparison

Capability	LangGraph	AWS Bedrock AgentCore	Itential	Mythos Advantage
Architecture Paradigm	Code-first state machine library.	Managed, opaque cloud runtime.	Low-code visual workflow builder.	**LangGraph.** Allows formally verified, code-defined logic.
Durability & Crash Recovery	Poor (natively). Requires external wrapping (Temporal).	Excellent. Managed service handles state persistence.	Excellent. Native durable workflow engine.	**Bedrock / Itential.** LangGraph requires heavy custom engineering to survive crashes.
Authority Separation	N/A. The developer must build the policy and authorization engine/ deterministic executor boundary.	Poor. Tools executed via AWS Lambda with AWS roles.	Fair. RBAC built-in, but lacks cryptographic capability grants.	**LangGraph.** Allows strict a governed agent platform architecture implementation.
Vendor Lock-in	Low. Open-source Python. Runs anywhere.	Critical. 100% coupled to AWS APIs, models, and IAM.	High. Proprietary platform, visual logic is trapped.	**LangGraph.** Zero infrastructure lock-in.
Context Engineering	Manual. Developer must wire up vector DBs and memory.	Good. Native RAG integration via Bedrock Knowledge Bases.	Poor. Not designed for complex LLM context window management.	**Bedrock.** Turn-key RAG integration.
Target Use Case	Custom AI products, sovereign agent runtimes.	Cloud-native enterprise automations.	Network/IT infrastructure provisioning.	**Context Dependent.**

3. Deep Dive: LangGraph

Architectural Philosophy: "The Stateless Graph Library"

LangGraph extends LangChain by introducing cycles and statefulness to LLM workflows. It treats the agent as a directed graph where nodes are functions (LLM calls, tool calls) and the state is passed along the edges. It is a library, not a runtime.

Pros

- **Ultimate Flexibility:** You own the cognitive loop. You can implement any architecture, including the strict a governed agent platform separation of planning and workflow engine (planner) and deterministic executor (executor).
- **Zero Vendor Lock-in:** Python code that runs on your laptop, a bare-metal server, or an air-gapped sovereign datacenter. You choose the model (local vLLM or hosted API).
- **Version Control:** Because it is pure code, agent logic can be peer-reviewed, tested in CI/CD, and formally verified.

Cons

- **The Durability Gap:** Natively, LangGraph is in-memory. If the Python process crashes mid-workflow, the state is lost. Building a crash-proof production agent requires wrapping LangGraph in a durable runtime like Temporal or DBOS.
- **The Security Burden:** LangGraph provides no native security boundaries. If you connect an LLM directly to a tool, a prompt injection can execute arbitrary code. You must build the policy and authorization engine policy engine yourself.
- **Infrastructure Overhead:** You are responsible for scaling, load balancing, and observability.

Best Use Case

LangGraph is the ideal choice for organizations building **proprietary AI products or sovereign runtimes (like a governed agent platform)**. If the engineering team has the maturity to wrap LangGraph in a durable execution runtime and implement zero-trust capability boundaries, it provides the ultimate foundation with zero lock-in.

4. Deep Dive: AWS Bedrock AgentCore

Architectural Philosophy: "The Turn-Key Cloud Runtime"

Bedrock AgentCore abstracts away the complexity of agent orchestration. You define an agent, connect it to Knowledge Bases (RAG), and define Action Groups (Lambda functions for tools). AWS handles the reasoning loop, tool invocation, and state management.

Pros

- **Instant Production Readiness:** Zero infrastructure to manage. AWS handles scaling, high availability, and crash recovery natively.
- **Seamless Ecosystem Integration:** Natively integrates with AWS IAM, S3, Lambda, and the Bedrock model garden.
- **Managed RAG:** Bedrock Knowledge Bases handle vector chunking, embedding, and retrieval without requiring the user to manage a Pinecone/pgvector cluster.

Cons

- **Critical Vendor Lock-in:** The agent logic, memory, and tools are deeply coupled to AWS APIs. Migrating this to Azure or a self-hosted runtime requires a complete rewrite.
- **Authority Boundary Failure:** Tools in Bedrock are executed via Lambda functions using AWS IAM roles. If the agent is compromised, it executes with the permissions of the Lambda role, violating the Mythos mandate that the vendor (AWS) must not hold the execution authority.
- **Opaque Cognitive Traces:** While AWS provides trace events, the deep cognitive telemetry (OpenTelemetry semantic conventions) is limited. You are relying on AWS's word for what the agent did.

Best Use Case

Bedrock AgentCore is the ideal choice for **AWS-native enterprises prioritizing speed-to-market over architectural sovereignty**. If the organization is already 100% committed to AWS, does not require air-gapping, and accepts AWS IAM as the root authority boundary, Bedrock provides the fastest path to production.

---

5. Deep Dive: Itential

Architectural Philosophy: "Visual Infrastructure Orchestration"

Itential is an enterprise automation platform designed to bridge the gap between IT, Network, and Cloud. It uses a visual builder (drag-and-drop nodes) to create workflows that configure routers, firewalls, and cloud APIs. It recently integrated AI to allow natural language to trigger these workflows.

Pros

- **Massive Adapter Catalog:** Itential has pre-built API adapters for thousands of network devices (Cisco, Juniper) and SaaS platforms, drastically reducing integration time.
- **Deterministic Durability:** Built on a robust workflow engine, it survives crashes and retries failed steps flawlessly.
- **Enterprise RBAC:** Granular role-based access control suited for large Network Operations Center (NOC) teams.

Cons

- **Not a Cognitive Runtime:** Itential is a deterministic workflow engine with an LLM bolted on. The AI is used to *select* a workflow, not to reason through a multi-step, probabilistic problem.
- **Visual Builder Liability:** Complex logic built in a drag-and-drop UI becomes "spaghetti" that cannot be peer-reviewed in Git, formally verified, or easily tested.
- **Missing Agent Features:** Lacks native LLM context engineering, memory graphs, or semantic UI protocols. It is not built for autonomous, multi-step AI reasoning.

Best Use Case

Itential is the ideal choice for **traditional Network/IT Operations teams** looking to automate complex, multi-vendor infrastructure provisioning. If the goal is to use AI to trigger deterministic network configurations via a UI, Itential excels. It is not suitable as a cognitive agentic runtime.

---

6. Alignment with Mythos Standards (MYTHOS-AI-0001 & MYTHOS-PLT-0004)

The Mythos Standard mandates three absolute requirements for agentic frameworks:

1. **Authority Separation:** The reasoning engine must be strictly isolated from tool execution via cryptographic policy.
2. **Durability:** Workflows must survive process crashes and resume exactly where they left off.
3. **Escape Hatches:** The framework must not create irrevocable vendor lock-in.

How LangGraph Scores: 2.5 / 5.0 (Natively) -> 5.0 / 5.0 (When wrapped in a governed agent platform) Natively, LangGraph fails the durability and security mandates. However, because it is pure, flexible code, it allows an organization to build the a governed agent platform architecture around it. When paired with Temporal (for evidence and history store/durability) and OPA (for policy and authorization engine/security), it perfectly achieves the Mythos standard.

How AWS Bedrock Scores: 3.5 / 5.0 Bedrock passes the durability mandate flawlessly. However, it fails the authority separation mandate (AWS IAM holds the keys) and critically fails the escape hatch mandate. The vendor lock-in is absolute.

How Itential Scores: 2.0 / 5.0 Itential passes the durability mandate but fails the core agentic requirements. It is a workflow engine, not a cognitive runtime, and its visual builder violates the Mythos mandate that all logic must be version-controlled code.

7. The Verdict

These three platforms serve entirely different architectural needs. Do not mistake them for direct substitutes.

Choose Itential if: Your primary goal is network/IT infrastructure automation. It is a configuration management tool with AI features, not an autonomous agent framework. *(Best for NOC/NetOps)*

Choose AWS Bedrock AgentCore if: Your organization has accepted total AWS dependency and prioritizes instant time-to-value over architectural sovereignty. Be prepared to rewrite everything if you ever leave AWS. *(Best for Cloud-Native Enterprise)*

Choose LangGraph if: You are building a proprietary, sovereign AI capability. If your engineering team has the maturity to implement durable execution (Temporal) and zero-trust boundaries (OPA/WASM), LangGraph provides the ultimate, lock-in-free foundation. *(Best for Sovereign AI / Custom Products)*

Mythos Recommendation: For organizations building the a governed agent platform architecture, **LangGraph (wrapped in a Temporal/OPA control plane)** is the mandated choice. It is the only framework that allows the strict implementation of the Mythos Agentic Standard, ensuring the agent remains durable, governed, and completely portable across any infrastructure.

```
LangGraph builds the brain.
Bedrock rents the brain.
Itential wires the nerves.

A managed runtime is a lease.
A visual builder is a liability.
A code-first library is an asset.

> Agents may be probabilistic.
> Architectural sovereignty must be absolute.
```

End of Evaluation CMP-004

CURRENT AUTHORITY REGISTER

External authority and freshness

This candidate publication uses the following current or explicitly rolling authorities as directional reference points. Their inclusion does not establish certification, legal equivalence, or implementation effectiveness. Exact editions and applicability must be revalidated before formal conformance claims.

Authority	Publication	Status	Use in this library
NIST-AI-RMF-1.0 NIST	Artificial Intelligence Risk Management Framework (AI RMF 1.0)	Current voluntary framework	AI risk governance, trustworthiness, measurement, and management.
NIST-AI-600-1 NIST	Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile	Final	Generative-AI risks and controls across design, development, deployment, and use.
NIST-AI-100-2e2025 NIST	Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations	Final with published planning note	Adversarial ML terminology, attack lifecycle, mitigations, and limitations.
NIST-SP-800-218A NIST	Secure Software Development Practices for Generative AI and Dual-Use Foundation Models	Final	Secure AI model development and supply-chain practices.
NIST-SP-800-63-4 NIST	Digital Identity Guidelines, Revision 4	Final	Identity proofing, authentication, federation, privacy, and usability.
NIST-SP-800-226 NIST	Guidelines for Evaluating Differential Privacy Guarantees	Final	Evaluation of differential-privacy guarantees and implementation claims.
NIST-PRIVACY-FRAMEWORK NIST	NIST Privacy Framework	Version 1.1 program page; exact release artifact must be pinned before conformance claims	Enterprise privacy-risk governance and data processing outcomes.
OWASP-LLM-TOP10-2025 OWASP GenAI Security Project	OWASP Top 10 for LLM Applications 2025	Current published awareness list	LLM application threat classes and mitigation guidance.
OWASP-AGENTIC-TOP10-2026 OWASP GenAI Security Project	OWASP Top 10 for Agentic Applications	Published; rapidly evolving	Autonomous-agent risks, tool misuse, memory compromise, and authority failures.
OTEL-SEMCONV-1.43 OpenTelemetry	OpenTelemetry Semantic Conventions 1.43.0 and GenAI Semantic Conventions	Mixed stability; GenAI conventions under active development	Telemetry names, attributes, traces, metrics, and events for AI and agent systems.
W3C-VC-2.0 W3C	Verifiable Credentials Data Model 2.0	Recommendation	Machine-verifiable credentials, issuers, holders, verifiers, and tamper resistance.
W3C-DID-1.0 W3C	Decentralized Identifiers (DIDs) v1.0	Recommendation; DID 1.1 remains candidate work	Decentralized identifier syntax, data model, resolution, and verification methods.

Validated 2026-07-24. Rapidly changing specifications, model-provider behavior, and security guidance require event-triggered review in addition to the scheduled review date.