The TITANS Project: Architecture and Implementation

The TITANS Project: Architecture and Implementation

Abstract:
The TITANS (Technology for Intelligent Transformative Agents and Novel Self-Improvement) project aims to design and implement a cognitive architecture that supports curiosity-driven, self-improving agents. It integrates modules for perception, memory, abstraction, reasoning and decision making, connected through cross-modal attention and generative replay. This paper summarizes the current state of the project, describing each module, the agent interface, and our roadmap toward production readiness.

1. Introduction
The vision of TITANS is to build an artificial agent that autonomously learns from experience, stores and abstracts knowledge, and interacts naturally with users and external models. The project synthesizes ideas from convolutional capsule networks, variational autoencoders, graph neural networks, reinforcement learning and natural language interfaces. It comprises a research effort and an engineering roadmap.

2. Core Architecture (Milestones M1–M5)
The architecture is divided into five milestones:
- M1 – Perception. A capsule network (CapsNet) with cross-modal attention processes raw sensory input. This network converts multimodal stimuli into hierarchical embeddings.
- M2 – Long-Term Memory. A memory module encodes episodic traces and stores them in a key–value store. Memories are organized chronologically and semantically and are referenced by context and saliency.
- M3 – Generative Replay. A variational autoencoder learns a world model of the agent's environment. Generative replay allows the agent to reimagine past experiences and avoid catastrophic forgetting.
- M4 – Abstraction & Reasoning. An abstraction network built on a Transformer encodes knowledge into symbolic graphs. A graph attention network performs relational reasoning, enabling the agent to infer causal relations and plan.
- M5 – Agentic Core. The core uses an actor–critic reinforcement learning algorithm that balances exploration and exploitation. A Bayesian value network outputs uncertainty estimates; the reward function encourages epistemic actions.

3. Agent Interface and Workspace
The agent is exposed through a workspace that integrates a natural language interface and a REST API. Multiple agents can run concurrently, with connectors to external large language models via an OpenRouter proxy. The interface described in TMA_Agent_Interface supports context persistence, modular instantiation and user-level I/O. LivePortrait and diffusers generate visual representations of the agent state. The runtime is Python 3.11, accessible via CLI and REST.

4. Security, IP and Compliance
The security layer addresses supply-chain attacks by pinning dependencies, scanning packages with Trivy and hosting private mirrors of PyPI. Data privacy threats, such as

membership inference and model inversion, are mitigated via differential privacy during training. The project specifies a dead man’s switch for intellectual property: encrypted repositories and secret sharing protect critical assets. Compliance with the EU AI Act requires risk assessments, model cards and auditing mechanisms.

5. Current Status and Technology Readiness

Our team has produced detailed design documents and prototype implementations. The core modules (M1–M5) are implemented in Python using PyTorch and PyG and have been validated in isolation (Technology Readiness Level 4). The natural language interface and security layer are specified but not yet implemented (TRL 3). The IP protection procedures are documented but lack automation. The next phase includes building a chat API, establishing continuous integration and security hardening pipelines, generating compliance documents, and conducting a pilot with external users.

6. Future Work

The coming sprints will focus on:

- Implementing and evaluating the natural language interface using fine-tuned transformer models, with a web endpoint for chat.
- Creating a DevOps pipeline with unit tests, vulnerability scans and a software bill of materials.
- Producing compliance artefacts such as risk tables, model cards and Open Policy Agent rules.
- Deploying a pilot instance with telemetry to measure latency, accuracy and user satisfaction.
- Engaging with regulators and the public to ensure trust and alignment.

7. Conclusion

TITANS is an ambitious exploration of cognitive architectures that aim for self-improvement through curiosity and structured reasoning. While the conceptual framework is complete and prototypes exist, significant engineering work remains to transition from laboratory validation to a secure, compliant and socially accepted system.