Secure Distributed Multi-Agent Control Architecture for Humanoid Robots Using an Epistemic Blackboard and Local Blockchain Consensus

Aslan Alwi*, Munirah, Almudaya Research Institute Team
aslan.alwi@umpo.ac.id, munirah@umpo.ac.id

Independent Researcher at Almudaya Research Institute
Department of Informatics Engineering, Faculty of Engineering, Universitas Muhammadiyah Ponorogo, Indonesia

November 2025

Abstract

Humanoid robots are evolving toward increasingly sophisticated architectures where centralized control methods based on classical kinematics and dynamics are no longer adequate. As modern AI-driven humanoids incorporate higher degrees of freedom, multimodal perception, and adaptive behaviors, new paradigms of distributed intelligence and secure coordination are required.

This paper introduces a secure distributed multi-agent control framework in which every joint of a humanoid robot is conceptualized as a semi-autonomous agent with local decision-making capabilities. Coordination among agents is achieved through an *Epistemic Blackboard*, a shared semantic substrate that functions as a global knowledge space. To ensure the integrity, authenticity, and consistency of blackboard updates, we propose embedding a lightweight blockchain mechanism directly inside the robot. This internal cryptographic ledger enforces tamper resistance, prevents malicious manipulation, and validates multi-agent interactions in real time.

The resulting architecture integrates distributed control, hierarchical coordination, and blockchain-based security into a unified framework suitable for next-generation humanoid robots. This paper establishes the foundational concepts for a more advanced swarm-based humanoid architecture, developed further in a companion paper.

1 Introduction

Humanoid robots are entering a new transformative phase in which the traditional paradigms of centralized control, analytical kinematics, and rigid dynamic equations are no longer sufficient to meet the requirements of real-world operation. As contemporary humanoid platforms evolve toward higher degrees of freedom, richer sensor modalities, and AI-native behavioral capabilities—including compact real-time depth and stereo processing on embedded GPUs—they demand architectures that are fundamentally more distributed, adaptive, and semantically expressive than their predecessors.

In this emerging perspective, a humanoid robot can be conceptualized not merely as a mechanical structure directed by a top-level controller, but instead as a *distributed multi-agent system* embedded within a physical embodiment. Prior work in whole-body humanoid control motivates this perspective: low-level motor behaviors combined through hierarchical coordination [1], distributed real-time control loops [2], and multi-agent coordination for locomotion and whole-body behavior [3, 4].

At the lowest level of abstraction, each joint or actuator possesses local sensory inputs, computational processes, and control objectives. Rather than functioning as passive mechanical components, these joints operate as *semi-autonomous agents* capable of making independent decisions, responding to local disturbances, and contributing cooperatively to the global behavior of the robot [5, 6, 7].

However, the introduction of distributed autonomy raises several fundamental questions:

- How do these independent joint agents maintain coherent whole-body coordination?
- How do higher-level controllers integrate or override local decisions when necessary?
- How is shared knowledge represented, exchanged, and validated among hierarchical agent layers?
- How can we ensure that internal coordination remains secure, consistent, and resilient to faults or adversarial interference?

To address these challenges, this paper introduces the concept of an *Epistemic Blackboard* as the central knowledge substrate that allows multiple agents—ranging from joint-level controllers to task-level planners—to access, contribute to, and update a unified representational space. The conceptual foundation of blackboard-based coordination originates in early cognitive systems research [8].

Yet, the centralization of knowledge also becomes a critical vulnerability. If an attacker, faulty module, or compromised internal process were to corrupt the blackboard, the entire robot could exhibit unsafe, unstable, or malicious behavior. Therefore, internal cognitive security must be treated as a first-class requirement for humanoid systems operating in human environments.

To this end, we propose embedding a lightweight, locally maintained *internal blockchain layer* inside the humanoid robot. Existing work demonstrates the benefits of blockchain for multi-agent coordination and CPS security [9, 10, 11, 12], but no prior approach embeds the blockchain directly *inside a single embodied humanoid robot*.

The contributions of this paper are summarized as follows:

1. We formalize humanoid robots as distributed multi-agent systems composed of semi-autonomous joint-level agents and hierarchical coordination layers.
2. We introduce an Epistemic Blackboard as a shared semantic substrate that enables coherent multi-agent communication and decision-making.
3. We propose a local blockchain mechanism that secures blackboard updates, enforces internal integrity, and prevents unauthorized state manipulation.
4. We present an integrated architecture that combines epistemic coordination with cryptographic consensus to achieve secure, scalable humanoid control.

This paper serves as the foundational framework for a more advanced swarm-based humanoid architecture, which will be developed in a subsequent companion paper.

2 Background and Related Work

The proposed architecture builds upon several foundational areas in robotics, distributed artificial intelligence, and secure cyber-physical systems. This section reviews the relevant literature across four domains: distributed multi-agent systems, blackboard architectures, security challenges in humanoid robotics, and the application of blockchain technologies within cyber-physical and multi-agent environments.

2.1 Distributed Multi-Agent Systems

Distributed multi-agent systems (MAS) provide a conceptual and mathematical framework for modeling collections of autonomous or semi-autonomous entities that operate concurrently, share partial information, and coordinate to achieve global objectives. Classical MAS research has explored negotiation protocols, decentralized control, cooperative behaviors, and emergent computation.

Distributed control principles have been explored in multi-robot systems and formation control [5]. The idea of distributing control inside a single robot is supported by advances in real-time whole-body control [2] and multi-agent reinforcement learning for locomotion [3]. Classical whole-body control formulations for humanoids—such as operational space control [1] and online walking motion generation [4]—motivate the decomposition of a humanoid into interacting controllers.

In robotics, MAS principles have historically been applied to swarm robotics, multi-robot fleets, and distributed sensing networks. However, the application of MAS at the *intra-robotic* level—treating the internal components of a single robot as interacting agents—remains relatively underexplored. Humanoid robots, characterized by tens or even hundreds of degrees of freedom, are inherently suitable for such a decomposition, as each joint possesses localized sensing and actuation that can be harnessed for semi-autonomous decision making.

While prior work in distributed robot control emphasizes cooperation between spatially separated robots, the proposed approach treats components *within* one robot as agents embedded within a shared embodiment. This perspective introduces unique constraints related to embodiment, physical coupling, latency requirements, and safety that differ from classical MAS formulations.

2.2 Blackboard Architectures

Blackboard systems originated from early AI cognitive architectures [8]. In complex robots, variants of blackboard-like architectures have been used implicitly in real-time hierarchical control, especially in layered frameworks such as Stanford's control hierarchy [1]. The proposed *Epistemic Blackboard* extends this tradition into an explicit semantic substrate for distributed embodied agents.

In the context of robotics, blackboard-based designs have been used for sensor fusion, task planning, and hybrid deliberative–reactive systems. However, their adoption in real-time, embodied control frameworks has been limited due to the high temporal demands of physical interaction and motion control.

The *Epistemic Blackboard* proposed in this paper generalizes the classical concept by framing the blackboard not merely as a data store but as a shared *semantic substrate* in which distributed control agents negotiate intent, share body-state estimates, and align their local policies with global objectives. This epistemic dimension distinguishes the proposed approach from conventional blackboard usage.

2.3 Security Challenges in Humanoid Robotics

Security research in robotics has traditionally focused on cloud-connected systems, teleoperation channels, and network-layer vulnerabilities. While such concerns remain important, humanoid robots introduce an additional and critically underexamined dimension: the security of internal cognitive processes.

Humanoid robots require strong guarantees of internal cognitive integrity. Blockchain techniques have been proposed for securing robotic systems [9], multi-agent systems [10], and CPS infrastructures [11, 12]. However, existing deployments focus on inter-device networks, not intra-robot epistemic validation.

Current safety standards (ISO 10218, ISO/TS 15066, etc.) primarily address mechanical and sensor-level safety but do not provide guidelines for securing distributed cognitive architectures. This motivates the need for an internal security mechanism designed specifically for multi-agent humanoid control.

Modern humanoid locomotion and manipulation increasingly rely on hierarchical visuomotor systems [13, 14]. Real-time embedded perception (e.g., GPU-accelerated depth estimation) enhances the feasibility of decentralized decision-making inside a single robot. These advances complement the proposed distributed agent-based architecture.

2.4 Blockchain in Cyber-Physical and Multi-Agent Systems

Blockchain technology provides cryptographic integrity, immutability, and decentralized consensus mechanisms initially developed for financial applications. In recent years, blockchain has been explored within cyber-physical systems (CPS), intelligent transportation, IoT infrastructures, and distributed sensor networks. These studies demonstrate its utility for ensuring secure communication, task validation, and data provenance.

In multi-agent systems, blockchain has been proposed as a coordination mechanism that enforces honesty, enables verifiable interactions, and prevents tampering among agents. However, existing work overwhelmingly focuses on inter-device coordination across networked systems.

To the best of the authors' knowledge, no prior research has proposed embedding a *local, on-device blockchain* within a humanoid robot to secure internal cognitive coordination among distributed control agents. The approach introduced in this paper extends blockchain principles into the intra-robotic domain, providing integrity guarantees for epistemic updates and distributed decisions occurring within the robot's embodiment.

3 Conceptual Architecture

The proposed control framework conceptualizes a humanoid robot as a distributed cognitive organism composed of multiple layers of autonomous and semi-autonomous agents. These agents interact through a shared epistemic substrate known as the *Epistemic Blackboard*. This section outlines the structure and function of the three foundational components of the architecture: joint-level agents, hierarchical control agents, and the blackboard itself.

3.1 Joint-Level Agents

At the lowest level of the physical embodiment, each joint of the humanoid robot is represented as a *local agent* endowed with sensing, actuation, and computational capacities. Formally, each joint agent A_i is defined by:

- **Local State** s_i: joint angle, velocity, torque, temperature, and other sensor readings.
- **Local Policy** π_i: a control function (potentially a neural policy) that determines the torque or position command based on local state and blackboard context.
- **Local Objectives**: maintaining stability, avoiding mechanical limits, minimizing energy use, responding to disturbances.
- **Communication Channel**: interfaces for reading from and writing to the epistemic blackboard.

Joint-level agents operate at high frequencies (up to 1 kHz or higher), enabling rapid reflex-like reactions to perturbations while contributing to whole-body coordination. Unlike classical servo loops, these agents may incorporate adaptive control modules, learned policies, or context-dependent behaviors.

Each joint acts as a local agent with sensing, actuation, and computation. High-rate reflexive control at the joint level is consistent with prior approaches in adaptive and reflexive locomotion [6, 7]. Distributed torque control principles have been shown to improve robustness in complex humanoids [5].

3.2 Hierarchical Control Agents

Above the joint layer, additional agents operate at progressively higher levels of abstraction. These include:

- **Limb-Level Agents**: responsible for coordinating groups of joints (e.g., the arm, hand, or leg) to achieve local kinematic and dynamic goals.
- **Whole-Body Coordination Agents**: integrating balance, posture, and global movement patterns across multiple limbs.
- **Task-Level Agents**: responsible for semantic planning, goal selection, affordance reasoning, and interpreting high-level instructions.

Each hierarchical agent operates at a timescale appropriate to its semantic level. For example, task-level agents may update at 1–10 Hz, while whole-body agents operate around 50–200 Hz. All agents interact with one another indirectly through operations on the epistemic blackboard.

Hierarchical agents serve three primary roles:

1. **Aggregation**: combining partial states from multiple joint agents into coherent representations of limbs or the entire robot.
2. **Constraint Propagation**: enforcing safety boundaries, workspace constraints, and global objectives that local joint agents must follow.
3. **Semantic Guidance**: translating task-level intent into distributed control signals that modulate lower-level policies.

This layered organization provides both modularity and scalability while preserving the benefits of distributed autonomy.

Higher-level agents aggregate joint-level behaviors into limb coordination, whole-body balance [4, 15], and task-level planning [1]. This layered approach aligns naturally with the distributed paradigm.

3.3 The Epistemic Blackboard

The *Epistemic Blackboard* is the core shared knowledge substrate that enables distributed coordination. It is not merely a memory buffer; rather, it is a semantic space designed to represent, structure, and validate the knowledge required for coherent humanoid behavior.

The blackboard serves as the authoritative representational space for distributed agents. Its epistemic nature echoes classical cognitive system practices [8] but enhanced with secure update validation.

Conceptually, the blackboard contains:

- **Global Body-State Estimates**: fused kinematic, dynamic, and inertial representations of the robot's posture and motion.
- **World Model Information**: object locations, environment geometry, affordances, and predicted interactions.
- **Task-Level Intent**: high-level goals, planned trajectories, semantic objectives.
- **Constraint Representations**: joint limits, safety constraints, collision boundaries.

- **Agent Contributions**: proposals, corrections, and commitments from joint-level and hierarchical agents.

Every update to the blackboard is a structured epistemic act: an agent asserts a piece of knowledge, proposes an action, or modifies an existing representation. Because all agents depend on this shared substrate, maintaining its consistency and integrity is crucial.

The use of an epistemic rather than purely informational blackboard emphasizes the intentional, interpretative, and context-dependent nature of the stored representations. This semantic layering is essential for enabling multi-agent coordination within an embodied, safety-critical system like a humanoid robot.

4 Threat Model and Security Requirements

The introduction of a distributed multi-agent architecture and a shared epistemic blackboard significantly increases the expressive power and adaptability of humanoid robots. However, it also introduces new classes of vulnerabilities that must be rigorously addressed to ensure safe and predictable operation, especially in human environments. This section defines the threat model relevant to the proposed architecture and formalizes the corresponding security requirements.

4.1 Threat Model

The threat model considers both internal and external adversaries, as well as unintentional faults arising from sensor errors, malfunctioning modules, or corrupted local policies. We categorize potential threats into four primary domains:

4.1.1 1. Malicious Manipulation of the Blackboard

An adversarial process, compromised software module, or corrupted local agent may attempt to:

- introduce false state representations,
- overwrite valid knowledge entries,
- inject unsafe task goals or motion proposals,
- manipulate constraint boundaries (e.g., joint limits, collision maps).

Given that the blackboard is the authoritative source of shared knowledge, any such manipulation may lead directly to harmful or unstable robot behavior.

4.1.2 2. Unauthorized Agent Actions

Local agents may produce inconsistent or unsafe outputs, either due to adversarial tampering or internal malfunction. This includes:

- issuing torque commands that violate global constraints,
- publishing epistemic updates without proper authorization,
- bypassing hierarchical validation mechanisms.

4.1.3 3. Inconsistency Among Distributed Agents

Distributed controllers may disagree or act on outdated blackboard information if:

- network latency induces asynchronous views of the global state,
- concurrent agents overwrite each other's contributions,
- partial failures break communication links.

Such inconsistencies may lead to contradictory or physically dangerous motion commands.

4.1.4 4. Compromised Hierarchical Controllers

Higher-level agents (e.g., whole-body coordinators or task planners) possess greater authority over global decisions. A successful compromise of these controllers could result in:

- misaligned task intentions,
- invalid world model updates,
- propagation of erroneous constraints to lower layers.

Ensuring the integrity of these controllers is critical for robot-wide safety.

4.2 Security Requirements

Based on the identified threats, the proposed humanoid control architecture must satisfy the following security requirements:

4.2.1 1. Integrity of Epistemic Updates

Every modification to the blackboard must be authenticated, tamper-resistant, and cryptographically verifiable. No agent should be able to alter shared knowledge without proper validation.

4.2.2 2. Agent Authentication and Authorization

Each agent must have a verifiable identity, and only authorized agents may publish specific classes of updates. Joint-level agents, for example, may not override whole-body constraints unless explicitly permitted.

4.2.3 3. Consistency Maintenance

Concurrent updates from multiple agents must be serialized, validated, and reconciled to maintain a coherent global state. The system must prevent race conditions and resolve conflicts deterministically.

4.2.4 4. Fault Tolerance and Fail-Safe Behavior

The architecture must detect and isolate faulty agents, sensor anomalies, or corrupted policies. When inconsistencies arise, the system should revert to a safe fallback representation rather than propagating erroneous states.

4.2.5 5. Traceability and Auditability

All epistemic actions must be logged in a structured, immutable record. This enables:

- post-incident analysis,
- verification of safety protocol compliance,
- accountability for agent decisions,
- debugging of distributed behaviors.

4.2.6 6. Real-Time Validation

Security mechanisms must operate under strict real-time constraints. Validation processes must not impose unacceptable latency on joint-level or whole-body control loops.

4.2.7 7. Isolation Against Adversarial Influence

The architecture must ensure that no single compromised agent can force the robot into unsafe states. Compromised modules must be isolated, and their epistemic contributions rejected or flagged automatically.

4.3 Security Assumptions

To establish a well-bounded threat model, the following assumptions are made:

- Physical access to internal hardware is restricted during operation.
- All communication occurs within the robot's local computational environment (no external network dependency).
- Sensor signals may be noisy but not arbitrarily spoofed at the hardware level.
- Cryptographic primitives are computationally secure under standard assumptions.

These assumptions provide a realistic foundation for securing the proposed multi-agent architecture under operational conditions.

5 Local Blockchain Layer for Robot Security

This section builds upon blockchain foundations for MAS integrity [10] and CPS security [11] but applies them uniquely to an intra-robot context.

To ensure the integrity, traceability, and authorization of epistemic updates within the humanoid robot, we introduce a lightweight, locally embedded blockchain mechanism. Unlike conventional blockchains designed for distributed networks of independent machines, the proposed blockchain operates entirely *within* the robot's computational architecture. Its purpose is not financial transaction processing, but the maintenance of a tamper-resistant, cryptographically verifiable ledger of all epistemic actions performed by the robot's distributed control agents.

This section presents the structural design of the local blockchain, the consensus mechanism tailored for intra-robotic operation, and the role of verification agents responsible for securing the epistemic blackboard.

5.1 Design Principles

The blockchain layer must satisfy three critical design requirements:

1. **Real-time operability**: Block validation must occur under strict timing constraints without introducing unacceptable latency in control loops.

2. **Minimal computational overhead**: The design must remain lightweight enough to operate on embedded processors without specialized hardware acceleration.

3. **High integrity with bounded trust**: Each contributing agent must be able to verify the authenticity of epistemic updates even if other agents malfunction or behave adversarially.

Accordingly, the architecture employs simplified block structures, deterministic consensus, and short commit cycles optimized for high-frequency control environments.

5.2 Block Structure

Each epistemic update submitted by an agent is encapsulated in a *block candidate*. The structure of a block B_t at time t is defined as:

$$B_t = \{\text{timestamp}, \text{agent\_id}, \text{update\_type}, \text{payload}, \text{prev\_hash}, \text{signature}\} \tag{1}$$

Where:

- **timestamp**: the update time in the robot's local clock.
- **agent_id**: a unique cryptographic identity associated with each agent (joint-level, limb-level, or hierarchical).
- **update_type**: category of epistemic action (e.g., state assertion, constraint update, task proposal).
- **payload**: structured data representing the update.

- **prev_hash**: hash pointer to the previous committed block.
- **signature**: digital signature of the submitting agent.

The block payload is designed to be minimalistic and domain-specific to avoid overhead. The entire blockchain typically resides in high-speed embedded memory, allowing constant-time access and validation.

5.3 Consensus Mechanism

Unlike traditional public blockchains, the intra-robot blockchain does not require trustless decentralized consensus. Instead, it uses a fast, deterministic consensus algorithm tailored to distributed control agents that operate under unified embodiment.

We adopt a modified form of **Byzantine Fault Tolerant (BFT)** consensus optimized for:

- small validator set (dozens, not thousands of nodes),
- predictable real-time operations,
- deterministic commit rules,
- extremely fast confirmation times (sub-millisecond).

The validator set consists of:

- hierarchical control agents (primary validators),
- selected limb-level agents (secondary validators),
- integrity monitors (dedicated safety agents).

Let the validator group be denoted by $V = \{v_1, v_2, \ldots, v_k\}$. For an epistemic update to be accepted into the blockchain, it must receive:

$$\text{Quorum}(B_t) \geq \lceil \frac{2k}{3} \rceil$$

This ensures the system can tolerate up to $\lfloor (k-1)/3 \rfloor$ faulty or malicious agents.

5.4 Block Generation and Commitment

The commit cycle proceeds in four steps:

1. **Proposal**: An agent creates a block candidate representing its epistemic update.
2. **Broadcast**: The candidate block is broadcast to the validator set.
3. **Validation**: Validators verify:
    - digital signature authenticity,

- consistency with epistemic constraints,
- absence of conflict with previous committed blocks,
- structural correctness of the payload.

4. **Commit**: Upon achieving quorum, the block is chained to the ledger and becomes the authoritative representation of the epistemic update.

Invalid or conflicting updates are automatically rejected, flagged, and potentially used as evidence of malfunction or malicious behavior.

5.5 Verification and Validation Agents

To maintain the integrity of the epistemic blackboard, certain agents are designated as *verification and validation agents*. Their responsibilities include:

- monitoring and validating block proposals,
- detecting anomalies or repetitive invalid submissions,
- isolating compromised agents or sensors,
- ensuring adherence to safety and constraint models,
- enforcing access control policies (e.g., which agents may update which fields).

These agents act as the immune system of the robot's cognitive architecture, preventing harmful epistemic mutations from propagating into the blackboard and ultimately influencing physical motion.

5.6 Advantages of Local Blockchain Integration

The local blockchain layer offers several key benefits:

- **Tamper Resistance**: Internal state transitions cannot be altered without violating hash-chain integrity.
- **Accountability**: All agent actions become auditable events.
- **Robust Fault Isolation**: Malfunctioning agents can be identified and isolated rapidly.
- **Distributed Trust**: No single agent can dominate the epistemic substrate.
- **Consistency Guarantees**: Conflicting updates are prevented from corrupting the blackboard.

This cryptographic foundation enables the safe operation of distributed humanoid control systems under extremely demanding real-time constraints.

6 Integration of the Blockchain with the Distributed Agent Architecture

The local blockchain layer and the distributed multi-agent control system must operate as a unified framework to ensure coherent, real-time, and secure humanoid behavior. This section explains how the blockchain interacts with joint-level agents, hierarchical control agents, and the epistemic blackboard. The integration is designed to preserve the autonomy and responsiveness of individual agents while enforcing global consistency and cryptographic security.

6.1 Overall Integration Flow

In the proposed architecture, every epistemic update produced by any agent must pass through the blockchain validation process before it becomes part of the authoritative blackboard state. This ensures that:

- all updates are authenticated,
- all updates are cryptographically validated,
- conflicting or malicious updates are rejected,
- every committed update is traceable and immutable.

The overall flow can be summarized in five steps:

1. **Local computation**: an agent generates a proposed epistemic update.
2. **Block formation**: the update is encoded as a block candidate.
3. **Consensus**: validators verify and vote on the block.
4. **Commit**: the validated block is appended to the local blockchain.
5. **Blackboard update**: the epistemic blackboard integrates the committed block into its shared state representation.

This pipeline ensures that no information reaches the blackboard without passing through the robot's internal cryptographic security layer.

6.2 Interaction with Joint-Level Agents

Joint-level agents operate at high frequency and require rapid access to the blackboard. To prevent blockchain latency from interrupting control loops, we separate:

- **read operations**, which are always permitted with negligible delay,
- **write operations**, which must pass blockchain validation.

Joint-level agents write only small, localized epistemic updates, such as:

- local torque estimates,
- disturbance detection,
- local posture corrections,
- energy minimization signals.

Because their proposals are simple and low in semantic complexity, block validation remains extremely fast, typically requiring only a subset of validators.

6.3 Interaction with Hierarchical Control Agents

Hierarchical agents (e.g., limb controllers, whole-body coordinators, task agents) generate higher-impact updates and therefore require stricter validation. Their updates include:

- posture adjustments affecting multiple joints,
- balance corrections,
- global constraint updates,
- task goal proposals,
- world model modifications.

To prevent unsafe propagation of high-level errors, hierarchical agent updates undergo the full validator quorum. Only after passing these checks are their proposals committed to the blockchain and reflected in the epistemic blackboard.

6.4 Blockchain as a Gatekeeper for Epistemic Consistency

The blockchain functions as a gatekeeper that serializes and validates distributed updates. Without this cryptographic layer, simultaneous writing from multiple agents could cause:

- contradictory commands,
- race conditions,
- unsafe motion plans,
- stale state propagation,
- unauthorized overwriting of critical data.

By enforcing strict verification prior to acceptance, the blockchain ensures that the blackboard always reflects a consistent and globally validated state of the robot.

6.5 Conflict Resolution and Update Ordering

Because multiple agents may attempt to modify overlapping parts of the blackboard, conflict resolution rules are enforced within the blockchain:

- **Temporal ordering**: blocks are serialized based on deterministic consensus rounds.
- **Hierarchical precedence**: higher-level agents override lower-level ones where appropriate.
- **Constraint protection**: epistemic entries that encode safety constraints cannot be overwritten without special multi-stage validation.
- **Rejection of unsafe proposals**: updates that violate physical, kinematic, or safety constraints are automatically rejected.

This ensures robust arbitration among agents with potentially competing intents.

6.6 Blackboard Synchronization

After a block is committed, the blackboard synchronizes with the blockchain ledger:

- the committed update is parsed into epistemic structures,
- redundant or derived entries are propagated,
- dependent modules are notified via event triggers,
- local caches are invalidated or refreshed.

This cycle ensures that all agents observe a synchronized and authoritative global state.

6.7 Operational Advantages of the Integrated Design

Integrating the blockchain with the distributed agent architecture yields multiple operational benefits:

- **Predictable Coordination**: all agents rely on a single validated source of truth.
- **Security and Integrity**: no update can bypass cryptographic validation.
- **Fault Isolation**: malicious or erroneous agents are detected and quarantined early.
- **Transparency**: all epistemic actions are recorded and auditable.
- **Real-time Stability**: the architecture supports deterministic, low-latency validation cycles.

This integration forms the computational backbone for secure and coherent humanoid control under distributed, multi-agent conditions.

7 Case Study: Distributed Control and Secure Coordination in a Balance Recovery Scenario

To illustrate the practical operation of the proposed architecture, this section presents a formal example scenario in which the humanoid robot experiences an external perturbation and must execute a rapid balance recovery maneuver. The scenario highlights the interactions among joint-level agents, hierarchical controllers, the epistemic blackboard, and the blockchain validation layer.

7.1 Scenario Description

Consider a humanoid robot standing upright when an unexpected lateral force is applied to its torso, causing a deviation in its center of mass (CoM). To maintain stability, the robot must coordinate a whole-body response involving multiple joints, limbs, and corrective policies. This requires:

- rapid sensing of perturbation forces,
- local reflex-like responses at disturbed joints,
- whole-body rebalancing strategies,
- consistent integration and validation of state updates,
- prevention of unsafe or contradictory corrective actions.

The distributed architecture ensures that these behaviors emerge coherently and securely.

7.2 Step-by-Step Operational Flow

The sequence of events unfolds as follows.

7.2.1 1. Perturbation Detection by Joint-Level Agents

Sensors at the hip, ankle, and torso joints detect:

- abnormal torque spikes,
- unexpected angular acceleration,
- deviation from predicted joint trajectories.

Each affected joint A_i generates a local epistemic update:

$$u_i = \text{“Disturbance detected: magnitude } m_i, \text{ direction } d_i''.$$

This update is encoded into a block candidate and sent to the blockchain validation layer.

7.2.2 2. Local Blockchain Validation

Validators (limb controllers + integrity monitors) verify:

- authenticity of each joint-agent identity,
- structural correctness of the update,
- consistency with physical constraints,
- absence of noisy or contradictory measurements.

Valid updates are committed to the blockchain and forwarded to the epistemic blackboard.

7.2.3 3. Blackboard Aggregation and Global Disturbance Estimation

The epistemic blackboard integrates the committed blocks and infers a global disturbance estimate through:

- fusion of distributed joint signals,
- inference of CoM displacement,
- estimation of required corrective momentum.

A global entry is created:

$$U_{\text{global}} = \text{“Lateral disturbance detected: estimated magnitude } M, \text{ direction } D''.$$

This entry will guide higher-level agents in the next step.

7.2.4 4. Whole-Body Controller Proposes a Rebalancing Plan

The whole-body controller (WBC), operating at an intermediate control frequency, generates a corrective motion plan:

$$P = \{\text{ankle shift, hip rotation, arm counter-swing}\}.$$

The proposal is sent to the blockchain as a high-level epistemic update requiring full quorum validation due to its global impact.

Validators check:

- consistency with joint torque limits,
- compliance with balance constraints,
- safety with respect to the environment.

If validated, the plan is appended to the blockchain.

7.2.5 5. Blackboard Update and Distributed Execution

The blackboard integrates the validated plan and issues structured directives for joint-level agents: Joint j: "Execute torque adjustment $\tau_j(t)$ under global plan P."

Joint-level agents then:

- compute local policies in the context of global corrective intent,
- coordinate implicitly through their shared blackboard view,
- apply torques that collectively realize the rebalancing maneuver.

7.2.6 6. Stability Restored and Epistemic Logging

Upon successful stabilization:

- joint agents submit "stability status" updates,
- the WBC posts a "balance restored" entry,
- all updates are committed to the blockchain,
- the robot's internal epistemic log now contains a fully traceable record of the incident.

This immutable record enables post-event analysis and safety verification.

7.3 Analysis of the Example Scenario

The scenario highlights several advantages of the proposed architecture:

- **Distributed responsiveness**: joint-level agents respond immediately to local disturbances.
- **Hierarchical coherence**: whole-body policies emerge naturally through blackboard coordination.
- **Security and integrity**: all updates are cryptographically validated before influencing behavior.
- **Conflict avoidance**: unsafe or inconsistent proposals are filtered before reaching the blackboard.
- **Auditability**: every epistemic action is permanently recorded.

This demonstrates how the robot maintains both physical safety and epistemic security under perturbations.

8 Discussion

The proposed architecture integrates distributed multi-agent control, epistemic coordination, and local blockchain security into a unified framework for next-generation humanoid robots. This section analyzes the conceptual implications, practical advantages, and inherent limitations of the approach, as well as its relationship to contemporary research directions in robotics and AI.

Connections to multi-agent control [3], operational whole-body robotics [4, 15], cognitive architectures [8], and blockchain-secured CPS [12, 11] are highlighted.

8.1 Conceptual Implications

Recasting a humanoid robot as a distributed cognitive system composed of semi-autonomous agents challenges traditional assumptions about robotic control. Classical architectures rely on centralized, deterministic motion planning and kinematic solvers, whereas the proposed model distributes intelligence across the embodiment itself. This shift aligns humanoid robotics more closely with principles from embodied cognition, cybernetics, and biological motor control.

The epistemic blackboard functions as a shared workspace for aligning agent intent, enabling coordination through a semantically structured medium rather than through direct message passing or rigid hierarchical commands. This approach supports a more flexible and context-aware form of collaboration among heterogeneous agents operating at different timescales and abstraction levels.

Embedding blockchain mechanisms within the robot introduces a new concept: *on-device epistemic security*. This transforms the blackboard into a tamper-resistant cognitive substrate in which every update is cryptographically validated and permanently recorded. Such a foundation strengthens trustworthiness and accountability within the robot's internal decision-making processes.

8.2 Practical Advantages

Several operational benefits emerge from the integration of distributed control with secure epistemic coordination:

- **Safety and Stability**: By preventing malformed or malicious epistemic updates, the architecture enhances physical stability, especially during high-risk motions such as balance recovery or manipulation of heavy objects.
- **Scalability**: The multi-agent decomposition allows the system to scale naturally to robots with increasing degrees of freedom or modular body structures.
- **Modularity**: New agents or behaviors can be integrated with minimal changes to the existing architecture, provided they adhere to validation rules.
- **Transparency and Auditability**: The blockchain provides a transparent, immutable record of the robot's internal reasoning and motor decisions.
- **Fault Tolerance**: Localized faults can be quickly detected and isolated before they escalate into whole-body failures.

Taken together, these properties enable humanoid robots to operate more safely and reliably in complex, dynamic environments.

8.3 Limitations

Despite its advantages, the architecture introduces several challenges and limitations:

- **Computational Overhead**: Even a lightweight blockchain imposes additional cycles for validation and state management. Ensuring real-time operation requires extremely efficient implementations.
- **Memory Footprint**: Maintaining an embedded ledger consumes memory resources, although pruning or snapshot techniques can mitigate long-term growth.
- **Agent Complexity**: Joint-level agents require greater computational capability than traditional servo loops, increasing design complexity at the hardware and firmware levels.
- **Design Trade-offs**: Stricter validation improves safety but may introduce latency, forcing trade-offs between responsiveness and security.
- **Formal Verification**: Ensuring correctness under all possible agent interactions remains a challenge for future work.

These limitations do not negate the feasibility of the architecture but highlight areas requiring careful engineering and optimization.

8.4 Position in the Broader Research Landscape

The architecture relates to and extends several major research areas:

- **Embodied Multi-Agent Systems**: It brings MAS principles inside a single embodied robot, an underexplored direction.
- **Cognitive Robotics**: The epistemic blackboard aligns with research on shared representation spaces in cognitive architectures.
- **Blockchain for CPS**: While blockchain has been studied for distributed systems, its application *inside* a single autonomous system is novel.
- **Humanoid Control**: The architecture complements modern AI-first humanoid approaches that rely on learned policies, by adding a secure semantic layer around them.

Thus, the framework not only proposes an engineering solution but also contributes a conceptual bridge between multi-agent AI, cognitive architectures, and secure robotic control.

8.5 Implications for Future Humanoid Designs

The proposed architecture paves the way for:

- more adaptive and self-organizing humanoid systems,
- enhanced security models that integrate physical and epistemic safety,

- integration with swarm-based and evolutionary control methods,
- modular robots with interchangeable agent-based control units,
- shared control across hybrid neural-mechanical architectures.

These implications form the conceptual foundation for the second paper, which extends the framework toward swarm-based control for humanoid robots.

9 Conclusion and Future Work

This paper presented a novel architecture for humanoid robot control that treats the robot as a distributed multi-agent system operating within a shared epistemic environment secured by a locally embedded blockchain. The integration of joint-level agents, hierarchical controllers, an epistemic blackboard, and a lightweight blockchain consensus mechanism provides a coherent and secure framework for managing complex, real-time, and safety-critical behaviors in humanoid robots.

By recasting joint actuators as semi-autonomous agents with local objectives and distributed responsibilities, the proposed architecture moves beyond classical centralized paradigms of robot control. The epistemic blackboard enables flexible, heterogeneous collaboration among agents at multiple abstraction levels, while the blockchain layer ensures that all epistemic updates are authenticated, immutable, and validated prior to influencing the robot's behavior.

The resulting framework strengthens the robustness, modularity, and transparency of humanoid control architectures, providing several advantages:

- enhanced safety through cryptographic validation of internal state transitions,
- improved scalability via distributed agent-based coordination,
- auditability and traceability of all internal decisions,
- resilience against internal faults and adversarial tampering,
- compatibility with AI-native and learning-based control policies.

While promising, the architecture also introduces computational overhead and requires efficient real-time implementations of both the blockchain and multi-agent coordination mechanisms. These challenges highlight opportunities for optimization and formal verification in future work.

9.1 Future Work

Several directions emerge naturally from this foundational architecture:

- **Swarm-Based Control Models**: Extending joint-level autonomy into swarm-like coordination dynamics, enabling emergent whole-body behaviors through distributed interaction rules.
- **Evolutionary and Adaptive Agents**: Enabling agents to adapt or evolve their policies over time while maintaining epistemic security.

- **Hybrid Neural-Blockchain Integration**: Leveraging neural networks to predict or validate blockchain updates in real-time, reducing computational cost.
- **Hardware Co-Design**: Developing actuator modules and embedded processors optimized for the demands of agent-based decentralized control.
- **Formal Epistemic Verification**: Employing logical frameworks to prove epistemic consistency and safety guarantees across distributed agent interactions.
- **Multi-Robot Extension**: Extending the architecture to fleets of humanoids that interact securely through inter-robot blackboards protected by shared blockchain layers.

The next stage of this research, presented in a companion paper, develops an extended model in which all joint-level agents are interpreted as elements of a coordinated swarm. This allows the robot to exhibit emergent, adaptive, and self-organizing motor behavior guided by swarm algorithms dynamically written onto the epistemic blackboard. Such an approach promises to unlock new capabilities in humanoid locomotion, manipulation, and embodied intelligence.

References

[1] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. Technical report, Stanford Robotics Lab / Tech. Report, 2005.

[2] Twan Koolen et al. Design of a distributed real-time control system for humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[3] Xiao Xiong et al. Distributed legged locomotion control using multi-agent reinforcement learning. *IEEE Robotics and Automation Letters*, 2023.

[4] André Herdt, H. Diedam, M. Diehl, et al. Online walking motion generation with automatic footstep placement. *Advanced Robotics / related venues (2010)*, 2010.

[5] Frank Romano et al. Distributed torque control for humanoid robots. In *IEEE International Conference on Robotics and Automation*, 2014.

[6] Mostafa Ajallooeian et al. Reflex-based locomotion control for biped robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2013.

[7] Jun Nakanishi et al. Learning from demonstration and adaptation of biped locomotion. *IEEE Transactions on Robotics*, 21(3):355–365, 2004.

[8] Barbara Hayes-Roth. Cognitive technologies: The design of joint human-machine cognitive systems. *AI Magazine*, 7(1):38–48, 1986.

[9] Ilya Afanasyev, Alexander Kolotov, Ruslan Rezin, Konstantin Danilov, Alexey Kashevnik, and Vladimir Jotsov. Blockchain solutions for multi-agent robotic systems: Related work and open questions. *arXiv preprint*, 2019.

[10] Ali Dorri et al. Blockchain in multi-agent systems: Consensus, security, and coordination. *arXiv preprint*, 2019.

[11] Tiago Fernandez-Carames and Paula Fraga-Lamas. A blockchain-based framework for secure multi-robot systems. *IEEE Access*, 8:183874–183888, 2020.

[12] Nikhil Kumar et al. Blockchain technology for cyber-physical systems: A review. *Computers & Electrical Engineering*, 75:697–717, 2019.

[13] Josh Merel et al. Hierarchical reinforcement learning for robot locomotion. *arXiv preprint*, 2019.

[14] He Zhang et al. Learning whole-body motor skills for humanoid robots. *arXiv preprint*, 2023.

[15] Adrien Escande, Abderrahmane Kheddar, and Michael Mistry. A survey of whole-body control techniques for humanoid robots. *IEEE Transactions on Robotics*, 32(4):738–752, 2016.