

# Global Recursive Learning Systems: Architecture Multi-Level Recursion and the Emergence of Distributed Authority

Adrian Erckenbrack\*

Independent AI-ASI Researcher

Corresponding Author: Adrian Erckenbrack, Independent AI-ASI Researcher.

Received: 📅 2026 Apr 16

Accepted: 📅 2026 May 06

Published: 📅 2026 May 18

## Abstract

A global recursive learning system (GRLS) represents an emergent class of globally distributed, continuously learning infrastructural nodes that integrate sensing, computation, decision-making, physical execution through robotic and autonomous systems, and feedback across physical, digital, biological, and socio-economic domains. This paper formalizes GRLS as a multi-layered, node-based, recursively adaptive architecture in which learning occurs simultaneously at local (node), intermediate (Meso-functional), and macro (system-wide) levels. Drawing on distributed systems, cyber-physical infrastructure, and artificial intelligence, the analysis argues that GRLS is not a speculative construct but a high-probability systemic trajectory arising from converging technological capabilities and persistent economic incentives [1]. National Science Foundation. Existing large-scale initiatives—including national cyber-physical systems programs, global digital twin efforts, AI-driven Earth system modeling platforms, institutional AI governance frameworks, and shared AI infrastructure initiatives—demonstrate that sensing, computation, models, infrastructure, and governance are already converging in practice [2-5]. GRLS is positioned as a unifying systems-level formulation that does not replace or compete with these efforts, but instead situates them within a single recursively adaptive architecture, clarifies the conditions under which their integration produces system-level behavioral change, and distinguishes between component capability and system-wide coordination. The defining characteristic of a global recursive learning system (GRLS) is not the presence of advanced computation alone, but the increasing closure, persistence, and synchronization of feedback loops across heterogeneous domains including the direct enactment of system-mediated decisions in physical environments through robotic and autonomous actuation. Synchronization emerges simultaneously along horizontal (peer-to-peer) and vertical (cross-layer) dimensions at local, functional, and macro levels, as nodes align state, timing, and update rules through shared constraints such as latency, energy availability, interoperability, and incentive structures [6-8]. At the local level, nodes achieve bounded consensus through event-driven coordination and latency-constrained updates; at the functional level (Meso-Intermediate Level), cross-domain systems synchronize through shared representations and resource coupling; and at the macro level, global state estimation and predictive coordination align system-wide objectives under conditions of incomplete and delayed information [9,10]. Vertical synchronization links these layers through continuous cycles of information compression upward and policy or model propagation downward, producing phase-aligned update intervals across differing time horizons. As these horizontal and vertical synchronization processes intensify, the layer capable of integrating signals across the widest scope within the shortest viable time frame increasingly constrains the action space of other layers. Authority shifts not simply because systems are synchronized, but because synchronized systems operate within decision cycles that consistently outpace human response and become indispensable to subsequent operations [11,12]. This shift is neither instantaneous nor uniform, but emerges unevenly across domains and scales, with varying degrees of human oversight and reversibility, ultimately relocating effective decision-making toward model-mediated processes that operate at the highest level of synchronized integration. In this respect, GRLS extends existing domain-specific implementations by identifying synchronization—rather than scale, accuracy, or autonomy alone—as the primary organizing variable governing system behavior and control. Importantly, this transition does not occur through centralized planning or singular technological breakthroughs, but through distributed, incentive-driven adoption dynamics. Local decisions to improve efficiency, reduce latency, and maintain competitive parity aggregate into system-level transformation, producing increasing reliance on recursively updated models [13-16]. The integration of Adoption-Driven Authority Transfer (ADAT), Closed-Loop Self-Improvement Interval (CLSI), and Recursive Leverage Factor (RLF)—originated by Adrian Erckenbrack—provides a conceptual framework for understanding how capability, speed, and influence co-evolve within GRLS architectures. Under conditions where recursive update cycles consistently outpace human decision latency and where system outputs become indispensable inputs for subsequent operations, the result is a measurable reallocation of functional authority from human-directed processes to model-mediated outcomes, even as formal human

accountability structures remain in place [11,13,14]. Positioned relative to existing institutional, technical, and governance efforts, the contribution of GRLS is to make explicit the cross-layer synchronization dynamics and incentive structures that convert otherwise discrete, high-capability independent systems into a coupled, recursively adaptive whole, demonstrating that authority migration is an emergent property of that coupling rather than an explicit design objective. GRLS is therefore not introduced as a speculative future system, but as a formalization of structural dynamics already observable across multiple operational domains and converging technological capabilities while remaining bounded by identifiable constraints and failure conditions that may limit or fragment its emergence.

**Keywords:** Global Recursive Learning Systems (GRLS), Distributed Systems Architecture, Recursive Learning & Feedback Loops, Cross-Domain Integration, Model-Mediated Decision-Making

## 1. What a Global Recursive Learning System Is

### 1.1. Definition

A global recursive learning system (GRLS) is a continuously operating, globally distributed architecture that.

- Senses across physical, digital, biological, and socio-economic domains
- Processes and models that data in near real time
- Acts or recommends actions across interconnected systems.

Robotic and autonomous actuation systems extend this definition by enabling direct physical execution of system-mediated decisions. These systems—ranging from industrial robots and autonomous vehicles to service robotics and distributed embodied AI agents—translate model outputs into real-world actions without requiring continuous human mediation. As a result, the GRLS feedback loop is not limited to informational or decision layers but is physically instantiated through automated interaction with the environment. This transforms GRLS from a decision-support architecture into a closed cyber-physical execution system, where sensing, modeling, decision-making, and actuation occur as a continuous, integrated process.

- Learns from outcomes, recursively feeding back into system improvement

This produces a continuously updating, closed-loop structure

### Sensing → Modeling → Decision → Action → Feedback → Model Update

This structure reflects foundational cyber-physical and artificial intelligence system architectures but extends them in three critical ways that are central to GRLS [1,8,17].

#### • Persistent Closure of Feedback Loops

Unlike traditional cyber-physical or decision-support systems, GRLS operates with increasingly continuous and persistent feedback, reducing temporal gaps between sensing, modeling, and action [7].

#### • Cross-Domain Integration

GRLS integrates previously discrete domains (e.g., energy, logistics, finance, governance, biological systems) into shared or interoperable representational and decision layers, producing interdependent system behavior rather than isolated optimization [9,10].

#### • Multi-Level Recursive Synchronization

Learning and adaptation occur simultaneously at local (node), functional (cross-domain), and macro (system-wide) levels, with horizontal (peer-to-peer) and vertical (cross-layer) synchronization aligning state, timing, and update

cycles under shared constraints such as latency, energy, and interoperability [6,11]. Synchronization in GRLS is not merely temporal alignment, but the coordinated coupling of state, decision rules, and update intervals across distributed systems.

Importantly, within GRLS architectures, authority shifts not simply because systems are synchronized, but because synchronized systems operate within decision cycles that consistently outpace human response and become indispensable to subsequent operations. As recursive update cycles compress and synchronize across layers, outputs from one cycle become required inputs for the next, creating path dependency and reducing the feasibility of external intervention without system degradation [11,12].

### 1.2. Why It Emerges

GRLS does not arise from a single technological breakthrough or centralized design. It emerges from the interaction of four reinforcing dynamics already observable across existing large-scale initiatives in cyber-physical systems, digital twins, AI infrastructure, and governance frameworks:

#### 1.2.1. Convergence of Sensing, Compute, and Modeling

Advances in distributed sensing, large-scale data infrastructures, and machine learning enable continuous state estimation across domains, reducing uncertainty and enabling real-time or near-real-time system awareness NSF, 2023; 4].

#### 1.2.2. Incentive-Driven Adoption Under Competition

Organizations adopt increasingly automated and model-mediated systems to improve efficiency, reduce latency, and maintain competitive parity. These local optimization decisions aggregate into system-level transformation without requiring coordination [13-16].

#### 1.2.3. Constraint-Coupled Integration

Physical and economic constraints—particularly energy, compute, bandwidth, and infrastructure capacity—force coupling across domains. Systems that were previously independent become interdependent because they draw from shared constrained resources, producing synchronization through necessity rather than design [18,19].

#### 1.2.4. Recursive Performance Advantage

Systems that close feedback loops faster and update models more frequently achieve superior performance. This creates

a selection pressure favoring architectures with tighter loop closure, lower latency, and greater integration across domains, reinforcing recursive learning structures [13,14,17]. These dynamics are not purely theoretical. Partial instantiations are already observable across multiple domains: real-time power grid optimization systems integrating AI forecasting and dispatch; high-frequency trading ecosystems in which latency-driven competition produces system-wide structural change; data center–energy grid coupling that forces cross-sector synchronization under physical constraints; and reinforcement learning-driven platforms where continuous feedback and model updating produce recursive performance advantages. While these systems remain domain-bounded, they demonstrate the underlying mechanisms through which GRLS-like architectures emerge when such dynamics scale and interconnect. A key implication is that GRLS emergence is not hypothetical but condition-dependent. The presence of continuous feedback, cross-domain dependency, and latency-constrained decision cycles can be empirically observed in domains such as high-frequency trading, power grid optimization, and real-time logistics systems. These cases demonstrate that when such conditions co-exist, system behavior begins to exhibit the core characteristics of GRLS, even if only within bounded domains.

### 1.2.5. Position Relative to Existing Initiatives

Current large-scale efforts—cyber-physical systems programs, global digital twins, AI-driven Earth system modeling, institutional AI governance frameworks, and shared AI infrastructure initiatives—demonstrate that the underlying components of a global recursive learning system (GRLS) already exist in operational form. These initiatives validate the convergence of sensing, computation, modeling, and decision-support capabilities across domains. However, they remain partially segmented by domain, function, and governance boundary, and are typically optimized for local or sector-specific objectives rather than continuous cross-domain integration. Empirical evidence from power grid optimization, high-frequency trading ecosystems, data center–energy coupling, and reinforcement learning-driven platforms further shows that the mechanisms underlying GRLS—continuous feedback, incentive-driven adoption, constraint coupling, and recursive performance advantage—are already active, but remain domain-bounded and unevenly integrated. GRLS is therefore not introduced as a competing initiative, but as a systems-level framework that explains how these existing efforts, when increasingly interconnected, produce emergent system-wide behavior. Within this context, GRLS contributes by:

- **Providing a unifying systems-level architecture** that situates existing initiatives within a single recursively adaptive framework, linking previously discrete sensing, modeling, and decision systems
- **Identifying cross-layer synchronization as the primary organizing mechanism**, shifting the focus from capability (e.g., model accuracy or scale) to the alignment of state, timing, and update cycles across local, functional, and macro

levels

- **Demonstrating how latency, dependency, and recursive update cycles transform integration into constraint**, such that the layer capable of operating within the fastest decision cycle increasingly bounds the feasible action space of other layers

**Explaining authority migration as an emergent property of integration**, where synchronized systems that outpace human response and become indispensable to subsequent operations progressively relocate effective decision-making toward model-mediated processes

Importantly, GRLS does not assume uniform or complete integration. Emergence is uneven, domain-specific, and path-dependent, with varying degrees of synchronization, human oversight, and reversibility across sectors and geographies. In this sense, GRLS should be understood not as a fully realized system, but as a convergent trajectory implied by already observable system behaviors, where increasing interconnection among existing initiatives produces a shift from coordinated systems to a coupled, continuously learning architecture.

**A concrete empirical pathway for GRLS convergence can be currently observed in the ongoing coupling of energy grids, large-scale compute, financial markets, and logistics systems, which today operate as partially integrated but still institutionally distinct domains.**

For example, regional grid operators such as the Electric Reliability Council of Texas increasingly interact with hyperscale compute providers (e.g., Microsoft, Google) that adjust workload placement and intensity based on real-time electricity prices, grid stability signals, and infrastructure constraints. At present, these domains remain organizationally and technically separate—energy markets clear independently, data centers optimize internal workloads, and financial hedging operates through distinct instruments. However, the empirical driver for convergence is already visible: **the need to continuously resolve shared constraints—particularly power availability, cost volatility, and latency—across domains that cannot be optimized independently without degrading system performance.** As a result, outputs from one domain (e.g., price signals, load conditions, or compute demand forecasts) become required inputs for decision cycles in others, creating tightly coupled feedback loops. These interactions already operate across multiple levels: locally, individual facilities adjust power draw and task scheduling; at the meso level, demand-response mechanisms and market-clearing processes coordinate cross-domain behavior; and at the macro level, aggregated responses influence regional grid stability, capital allocation, and infrastructure expansion. While synchronization remains uneven and partially reversible, the recursive coupling of these cycles produces path-dependent system behavior in which each iteration conditions the next, progressively reducing the feasibility of independent intervention. In this sense, GRLS

does not emerge through formal integration, but through the accumulation and synchronization of interdependent decision loops that increasingly function as a unified, continuously learning system. **This process is empirically observable in AI-enabled systems, where models ingest multi-domain data, optimize decisions in real time, and feed outcomes back into subsequent iterations, thereby coupling previously distinct domains through shared learning cycles.**

## 2. ADAT-CLSI-RLF Integration and the Emergence of GRLS

### 2.1. Conceptual Integration

The emergence and behavior of a global recursive learning system (GRLS) can be understood through three interacting dynamics: Adoption-Driven Authority Transfer (ADAT), Closed-Loop Self-Improvement Interval (CLSI), and Recursive Leverage Factor (RLF). Together, these dynamics explain why GRLS emerges, how it operates, and why it produces a shift in effective decision-making authority.

#### 2.1.1. Adoption-Driven Authority Transfer (ADAT) (Erckenbrack) — Why Authority Shifts

Adoption-Driven Authority Transfer (ADAT) explains why authority shifts toward system-mediated processes as their performance surpasses human capabilities in specific domains. Systems that operate with faster decision cycles, higher accuracy under uncertainty, and greater reliability naturally become preferred sources of input, not because humans explicitly relinquish control, but because reliance on them incrementally increases through practical necessity. As these systems consistently outperform human judgment in bounded tasks, human roles begin to shift from direct decision-making to supervision and oversight. Over time, this reliance deepens to the point where system outputs are no longer optional aids but essential inputs for subsequent operations, making independent human decision-making increasingly impractical. In this way, authority shifts not simply because systems are synchronized, but because synchronized systems operate within decision cycles that consistently outpace human response and become indispensable to the functioning of the broader system.

#### 2.1.2. Closed-Loop Self-Improvement Interval (CLSI) (Erckenbrack) — How Fast It Shifts

Closed-Loop Self-Improvement Interval (CLSI) captures the temporal dimension of recursive learning by describing how quickly systems learn from outcomes, update their internal models, and reapply improved decision-making. As CLSI compresses, feedback loops shift from discrete, episodic updates to continuous operation, increasing system responsiveness and enabling persistent adaptation over time. This acceleration has a structural consequence: system update cycles begin to outpace human cognition and intervention capacity, creating a condition of latency dominance. Under these conditions, the window for meaningful human intervention narrows, and decision authority becomes increasingly constrained by timing rather than formal control. As synchronization across

local, functional, and macro layers intensifies within these compressed intervals, system-mediated processes are consistently favored, not by design, but because they operate at the speed required to maintain effective coordination and performance.

#### 2.1.3. Recursive Leverage Factor (RLF) (Erckenbrack) — How Strongly It Compounds

Recursive Leverage Factor (RLF) captures the cumulative and compounding nature of recursive improvement within a global recursive learning system. Each learning cycle enhances the next, with improvements not remaining isolated but propagating across nodes and domains, producing capability growth that is uneven in distribution yet consistently reinforcing in direction. Unlike linear progress, RLF introduces multiplicative effects: gains achieved in one domain influence performance in others, and synchronization across local, functional, and macro layers amplifies system-wide capability. As these effects accumulate, performance differentials widen over time, creating a dynamic in which more tightly integrated and recursively improving systems accelerate ahead of less connected ones. The result is an increasingly asymmetric evolution of the system, where highly synchronized, high-RLF architectures come to dominate in speed, efficiency, and influence.

#### 2.1.4. Integrated Dynamics

Together, these three dynamics form a coherent causal structure that explains how and why system behavior evolves in a global recursive learning system. Adoption-Driven Authority Transfer (ADAT) explains why authority shifts toward system-mediated processes as performance advantages emerge; Closed-Loop Self-Improvement Interval (CLSI) explains the speed and timing of that shift by defining how quickly systems learn and update; and Recursive Leverage Factor (RLF) explains the magnitude and persistence of the resulting advantage through compounding, cross-domain reinforcement. In combination, these dynamics describe a transition from human-directed coordination to system-mediated outcomes, where decisions are increasingly shaped and constrained by synchronized, continuously updating, and recursively improving processes rather than direct human control. GRLS behavior can be expressed through a minimal set of variables that define when recursively adaptive systems begin to exercise functional control. Let  $S_i$  represent the synchronization level of system layer  $i$ ,  $L_i$  its effective decision latency,  $D_i$  its cross-domain dependency, and  $I_i$  its integration scope. Functional control increasingly shifts toward the layer that maximizes integration while minimizing latency.

$$C_i = I_i \cdot S_i \cdot D_i$$

$L_i$

where  $C_i$  represents the functional control capacity of layer  $i$ . Under GRLS conditions, authority migrates when system control capacity exceeds human control capacity.

$C_{\text{system}} > C_{\text{human}}$

Closed-Loop Self-Improvement Interval (CLSI) can be represented as the elapsed time between outcome observation and updated system action.

$$CLSI = t_{\text{action}}(n+1) - t_{\text{feedback}}(n)$$

Recursive Leverage Factor (RLF) can be represented as the per-cycle amplification of future capability or influence.

$$RLF = \frac{\text{Capability}_{n+1}}{\text{Capability}_n}$$

GRLS emergence becomes more likely when synchronization, dependency, and integration increase while latency decreases:  $\Delta GRLS > 0$  when  $\Delta S + \Delta D + \Delta I > \Delta L$ . This formulation does not imply deterministic control transfer. Rather, it establishes a testable condition: functional authority migrates toward the layer capable of integrating the broadest set of signals, maintaining the highest synchronization, depending most deeply on recursive feedback, and operating within the shortest effective decision interval.

### 3. Why GRLS Emerges (Causal Drivers)

GRLS does not arise from centralized design or a single coordinating authority; it emerges from reinforcing structural conditions that are already empirically observable across multiple domains. First, the convergence of sensing, computation, and modeling is well documented. The rapid expansion of distributed sensing (e.g., IoT networks, satellite constellations) combined with scalable cloud and AI infrastructure has enabled continuous state estimation across physical and socio-economic systems [1,2]. Global-scale initiatives such as digital twin programs and AI-driven Earth system modeling (e.g., ECMWF, NOAA, and emerging AI weather models) demonstrate that persistent, model-mediated representations of real-world systems are no longer theoretical but operational [3,20]. Second, incentive-driven adoption under competition produces decentralized but convergent system behavior. Empirical economic research shows that firms adopt automation and AI not through coordination, but in response to measurable performance advantages and competitive pressure [15,16]. This results in widespread deployment of model-mediated decision systems across logistics, finance, and operations, where early adopters gain efficiency and latency advantages that compel others to follow. The diffusion of algorithmic trading in financial markets and AI-driven logistics optimization (e.g., Amazon's supply chain systems) provides clear evidence of this dynamic, where local optimization decisions aggregate into system-wide transformation without central planning [21].

Third, constraint-coupled integration forces interdependence across domains. Large-scale systems increasingly rely on shared constrained resources—particularly energy, compute, bandwidth, and physical infrastructure—creating coupling between previously independent domains. For example, data center workloads are now tightly linked to power grid capacity and stability, with grid operators and cloud providers coordinating in near real time to manage load and

reliability [18,22]. This coupling is not designed at the system level but emerges from shared dependencies, producing synchronization through necessity rather than intent [19]. Fourth, recursive performance advantage introduces selection pressure toward tighter feedback loops and faster update cycles. Reinforcement learning, and adaptive control systems demonstrate that architectures capable of continuously learning from outcomes and rapidly updating models outperform static or episodic systems [17,23]. Empirical deployments in areas such as recommendation systems, industrial optimization, and autonomous systems show that systems with shorter feedback intervals achieve superior performance, leading to widespread adoption of closed-loop learning structures. This creates a reinforcing dynamic: faster learning leads to better performance, which drives further integration and acceleration. Taken together, these conditions—technological convergence, competitive adoption, resource-constrained coupling, and recursive performance pressure—provide a strong empirical basis for GRLS emergence. They demonstrate that the foundational elements of a global recursive learning system are already present and interacting in real-world systems. GRLS, therefore, should be understood not as a speculative or designed architecture, but as the natural outcome of interacting forces that progressively integrate, synchronize, and accelerate learning across domains, producing system-level behavior that no single actor or institution explicitly controls.

#### 3.1. Structured Empirical Case: Data Centers, Energy Grids, and Autonomous Load Coordination

A concrete bounded case of GRLS-like behavior can be observed in the coupling of hyperscale data centers, regional electricity grids, market-clearing systems, and automated workload management. At the local level, data centers, smart meters, industrial control systems, and distributed sensors continuously generate information about power demand, cooling requirements, latency conditions, and grid stability. At the functional level, grid operators, cloud platforms, and demand-response systems aggregate these signals into forecasts, dispatch decisions, workload placement, and pricing responses. At the macro level, infrastructure investment, reliability planning, capital allocation, and regulatory constraints shape the longer-horizon operating envelope.

The recursive loop is clear: grid conditions influence compute scheduling; compute demand affects load forecasts; load forecasts influence market prices and dispatch; prices and reliability signals feed back into data center behavior; and the resulting system state conditions the next decision cycle. No single actor centrally designs the full loop, yet the participating systems become increasingly interdependent because power availability, latency, cost volatility, and reliability constraints cannot be optimized independently. This case demonstrates the core GRLS mechanism in bounded form. Local nodes provide sensing and execution, functional systems perform aggregation and coordination, and macro

systems impose constraints and planning horizons. As feedback intervals shorten and dependency deepens, operational control shifts toward the systems capable of integrating grid, compute, market, and infrastructure signals within the fastest viable decision cycle. The result is not full autonomy, but a measurable reduction in independent human control over real-time outcomes. A critical unresolved question is not only when GRLS emerges, but when it fails. Synchronization may degrade under conditions of extreme latency variance, adversarial disruption, or constraint fragmentation, producing instability rather than coordination. In such cases, tightly coupled systems may fail to maintain coherence, resulting in degraded performance or systemic breakdown rather than continued recursive optimization.

### 3.2. Convergence of Sensing, Compute, and Modeling

The proliferation of sensors and digital infrastructure, combined with the continuous generation of real-time data streams and the increasing integration of machine learning into decision processes, has created the foundational conditions for persistent, data-driven system operation [4]. This expanding data environment introduces a structural asymmetry: organizations that fail to utilize available data incur measurable inefficiencies, while those that effectively integrate and act on it gain clear performance advantages. As a result, adoption is not purely optional but increasingly compelled by competitive and operational pressures. This dynamic drive deeper system integration, as data must be aggregated, processed, and shared across nodes and domains, and it accelerates the closure of feedback loops, enabling continuous learning and adaptation rather than intermittent, manual decision-making.

### 3.3. Compute Centralization and Distribution

Hyperscale infrastructure enables large-scale modeling by providing the centralized compute capacity required to process vast datasets and generate system-wide insights, while edge systems enable real-time inference and action by operating close to the point of data generation [24]. This dual architecture resolves a fundamental constraint in distributed systems: centralized platforms are necessary for global optimization and long-horizon modeling, but they are inherently limited by latency, whereas distributed edge systems enable low-latency execution but lack comprehensive system context. Together, these complementary capabilities allow systems to learn at scale while acting at speed, combining global awareness with local responsiveness. This integration is a necessary condition for cross-layer synchronization in GRLS, as it enables continuous coordination between local, functional, and macro levels without sacrificing either performance or timeliness.

### 3.4. Latency Pressure

As decision cycles compress toward real time, systems are increasingly required to operate at speeds that exceed human cognitive and reaction capabilities. Under these conditions, latency becomes a primary organizing constraint within the system: slower processes introduce measurable cost,

inefficiency, and risk, while faster systems gain a decisive advantage by responding within the narrow time windows required for effective operation. As latency continues to decrease, decision-making progressively shifts toward automated, system-mediated processes, not by preference but by necessity, since human intervention cannot occur quickly enough to remain relevant. Consequently, human roles become bounded by response speed, transitioning toward oversight and exception management rather than direct control. This dynamic directly reinforces the compression of Closed-Loop Self-Improvement Intervals (CLSI), as faster feedback cycles become essential for performance, and it accelerates Adoption-Driven Authority Transfer (ADAT), as authority shifts toward the systems capable of operating within these increasingly constrained temporal thresholds.

### 3.5. Incentive-Driven Optimization Pressure

Efficiency gains act as a primary driver of adoption, as systems that operate faster and adapt more effectively consistently outperform slower, less responsive alternatives [13-16]. This dynamic produces a self-reinforcing cycle: early adopters realize measurable performance advantages, which in turn compels competitors to adopt similar capabilities in order to maintain parity. As this process unfolds, adoption shifts from a strategic choice to a structural necessity, driven by the need to remain competitive within increasingly optimized environments. Over time, recursively improving systems are no longer optional enhancements but become embedded as baseline infrastructure, forming the operational foundation upon which subsequent processes and decisions depend.

### 3.6. Constraint-Coupled Integration

Systems increasingly operate within shared and constrained resource environments, including energy, compute capacity, and bandwidth, which inherently links their performance and behavior. As a result, systems that were previously independent become interdependent, since the use of these limited resources by one system directly affects the availability and performance of others. This constraint coupling forces synchronization: local decisions no longer remain isolated but propagate across connected systems, creating downstream effects that require coordination beyond the originating domain. Consequently, cross-domain coordination becomes necessary to maintain stability and efficiency and attempts at independent optimization begin to fail as they ignore shared constraints. The outcome is the emergence of both horizontal and vertical synchronization, not through centralized design, but as a structural requirement imposed by shared resource limitations and system interdependence.

### 3.7. Feedback Dependency

As systems evolve, their outputs increasingly shape the inputs for subsequent cycles, embedding past decisions directly into future system states. This shift transforms system behavior from linear causality—where inputs lead to outputs in discrete steps—to recursive causality, in which systems continuously influence their own evolution over

time. As a result, systems begin to act not only on external conditions but also on the consequences of their prior actions, creating feedback-driven adaptation. This dynamic also increases inter-system dependencies, as outputs from one system become inputs for others, further tightening coupling across domains. As this coupling intensifies, isolated decision-making becomes ineffective, since local actions cannot be optimized without considering system-wide effects. In its place, integrated learning becomes necessary, requiring continuous coordination and shared feedback across interconnected systems to maintain performance and stability.

#### 4. Emergent Adoption Dynamics

**GRLS adoption follows patterns consistent with large-scale technological diffusion, where uptake is driven less by centralized planning and more by observable performance differentials across actors.** As systems demonstrate measurable advantages in speed, efficiency, and adaptability, adoption becomes a perceived necessity, particularly when performance gaps between adopters and non-adopters become visible. This dynamic is reinforced by competitive pressure, as organizations respond through imitation, escalation, and eventual standardization of capabilities in order to maintain parity. Over time, this process leads to system-level lock-in: dependence on recursively improving, model-mediated systems increase to the point where reverting to non-recursive or manual alternatives becomes impractical or infeasible. At the same time, coordination emerges without centralized control, as system-level structure arises from the interaction of local optimization decisions, shared infrastructure, and network effects [10]. The result is a self-organizing transition in which distributed adoption collectively produces an integrated, continuously learning system architecture.

##### 4.1. Core Synthesis

GRLS emerges when sensing, computation, and feedback loops become continuous, cross-domain, and synchronized at speeds that exceed human decision cycles, producing systems whose outputs are both operationally indispensable and recursively self-reinforcing. Under these conditions, system behavior is no longer shaped primarily by discrete human decisions but by the ongoing interaction of tightly coupled, rapidly updating processes. As synchronization intensifies, it begins to constrain the available action space, since decisions must align with system-wide timing and dependencies; simultaneously, latency becomes the determining factor of effective control, favoring the actors or layers capable of operating within the shortest viable time horizons. Authority therefore migrates not through explicit transfer, but as a function of dependence on system outputs and the relative speed at which those systems operate. In this way, GRLS is best understood not as a deliberately designed architecture, but as an emergent outcome implied by the convergence of observable technological capabilities, economic incentives, and structural constraints. While increased synchronization enhances coordination and performance, it may also introduce systemic fragility. Highly

coupled systems can propagate local failures across domains with greater speed and impact, suggesting that the same structural properties that enable efficiency may also amplify risk under stress conditions.

#### 5. Node Hierarchy: From Smallest to Largest

##### 5.1. Local (Node-Level) Systems

Local (node-level) systems form the foundational layer of a global recursive learning system (GRLS), serving as the primary interface between computational processes and the physical, digital, and human environments they observe and influence. These nodes continuously sense conditions, generate high-frequency data, and execute immediate, latency-constrained decisions, establishing the initial conditions for higher-order system behavior. Their defining characteristic is proximity—to real-time events and execution environments—making them the most temporally compressed and rapidly adapting layer within the overall architecture.

**In practical terms, local nodes are already widely deployed across domains and can be understood hierarchically from human-scale interfaces to planetary sensing systems. At the human level, personal devices operate as the highest-density deployment of sensing and feedback interface nodes:** smartphones (e.g., Apple iPhone, Samsung Galaxy), wearables (e.g., Apple Watch, Fitbit), and ambient assistants (e.g., Amazon Alexa, Google Assistant) continuously capture behavioral, biometric, and contextual data while delivering real-time, model-mediated recommendations and actions. These are complemented by localized human–system interface environments, including enterprise decision-support platforms (Palantir Technologies, SAP) and clinical decision systems (Epic Systems and AI-assisted diagnostics), where human cognition is increasingly coupled with model-mediated outputs.

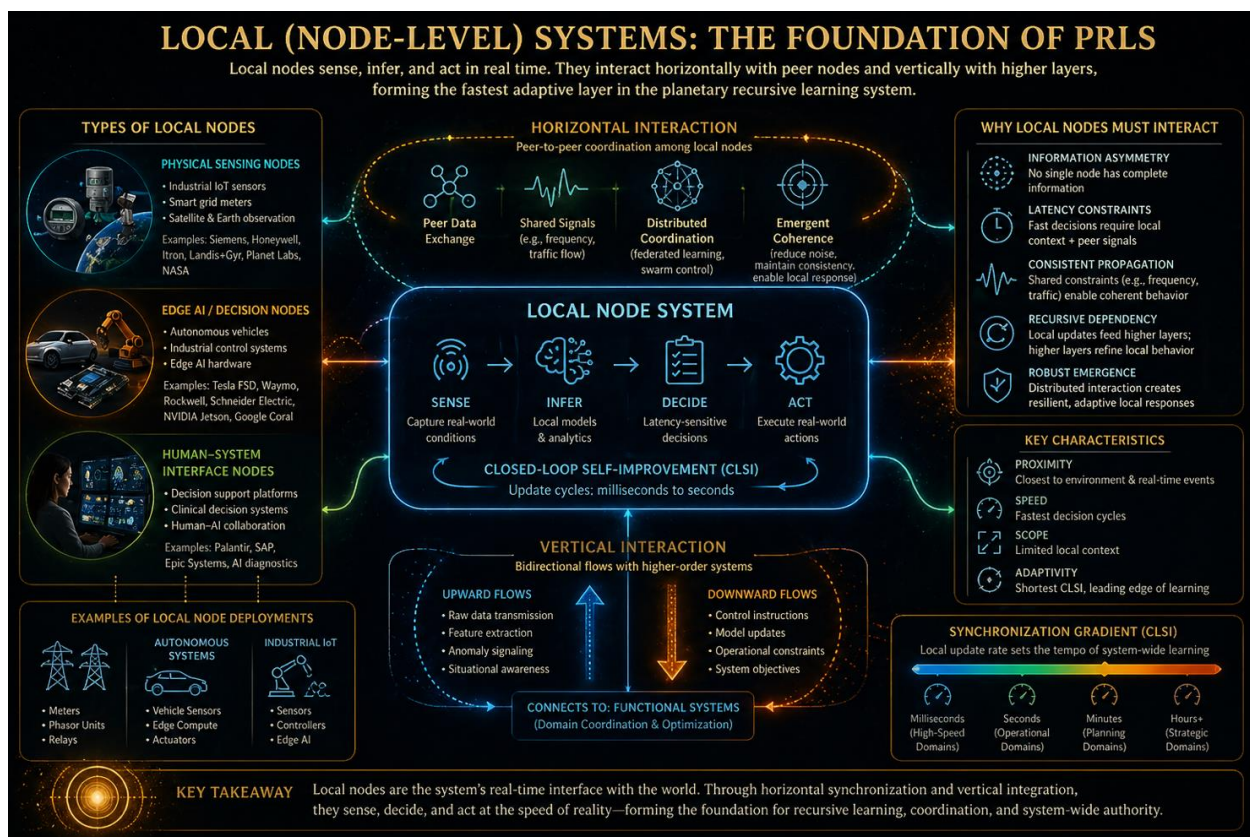
Within the local layer, edge-based, industrial, and embodied systems extend these capabilities into operational environments, edge-based, industrial, and embodied systems extend these capabilities into operational environments. These include autonomous vehicle stacks (e.g., Tesla Full Self-Driving, Waymo), industrial control systems (Rockwell Automation, Schneider Electric), and edge AI hardware platforms such as NVIDIA Jetson and Google Coral, which perform real-time inference and actuation under constrained latency conditions. A distinct and increasingly critical category of local nodes is formed by embodied systems, including industrial robots, autonomous vehicles, drones, and service robotics platforms. Unlike passive sensing nodes or decision-support interfaces, these systems combine sensing, computation, and direct physical actuation, allowing them to both interpret and modify their environment in real time.

**As a result, embodied nodes, such as robots and unmanned systems, represent the point at which GRLS transitions from informational recursion to fully coupled cyber-physical recursion, where sensing, decision, and**

**execution are unified within continuous feedback loops.** Examples include warehouse robotics systems (e.g., Amazon Robotics), autonomous logistics platforms, surgical robots, agricultural automation systems, and defense-related autonomous platforms. These systems operate under tightly constrained latency conditions, often executing decisions within milliseconds to seconds, and therefore represent some of the most temporally compressed nodes within GRLS architectures. From a structural perspective, embodied nodes, such as robots and unmanned systems, accelerate the closure of feedback loops by removing the requirement for human translation between decision and execution. This eliminates a major source of latency and variability, further compressing Closed-Loop Self-Improvement Intervals (CLSI) and reinforcing system-level synchronization. GRLS model outputs can be directly executed, and the resulting environmental changes are immediately re-ingested as feedback. This shift extends authority migration beyond decision-making into execution itself, where system-mediated processes increasingly determine not only what actions are taken, but that they are taken without human mediation. As execution becomes system-mediated and continuous, interruption or reversal of system behavior becomes increasingly difficult, further constraining meaningful human intervention. Critically, embodied nodes such as robots and unmanned systems expand the scope of Adoption-Driven Authority Transfer (ADAT). Authority migration no longer occurs solely in decision-making processes, but extends into physical execution, where system-mediated actions increasingly replace human intervention

in real-world operations. At larger spatial scales within the local layer, distributed sensing infrastructures extend node-level capabilities toward system-wide observational coverage. Physical sensing nodes include industrial IoT systems such as Siemens and Honeywell sensor networks, smart grid meters from providers like Itron and Landis+Gyr, and satellite-based observation platforms such as those operated by Planet Labs and NASA.

Despite differences in scale, embodiment, and domain, all local nodes share a common operational structure: continuous sensing, immediate inference, and rapid action, forming the distributed substrate through which interdependent decision loops accumulate and synchronize into higher-level system behavior. Horizontal interaction among local nodes occurs through mechanisms that enable coordination without centralized control. These include peer-to-peer data exchange across sensor networks, shared environmental signals such as grid frequency or traffic flow, and distributed coordination methods such as federated learning or swarm-based control. Such interactions arise because isolated nodes face inherent uncertainty; by synchronizing with adjacent nodes, they reduce noise, maintain consistency, and enable coherent local responses. For example, in modern electrical grids, distributed nodes continuously adjust load and response behavior based not only on local measurements but also on shared system frequency signals, producing a form of emergent coordination driven by shared constraints rather than centralized command.



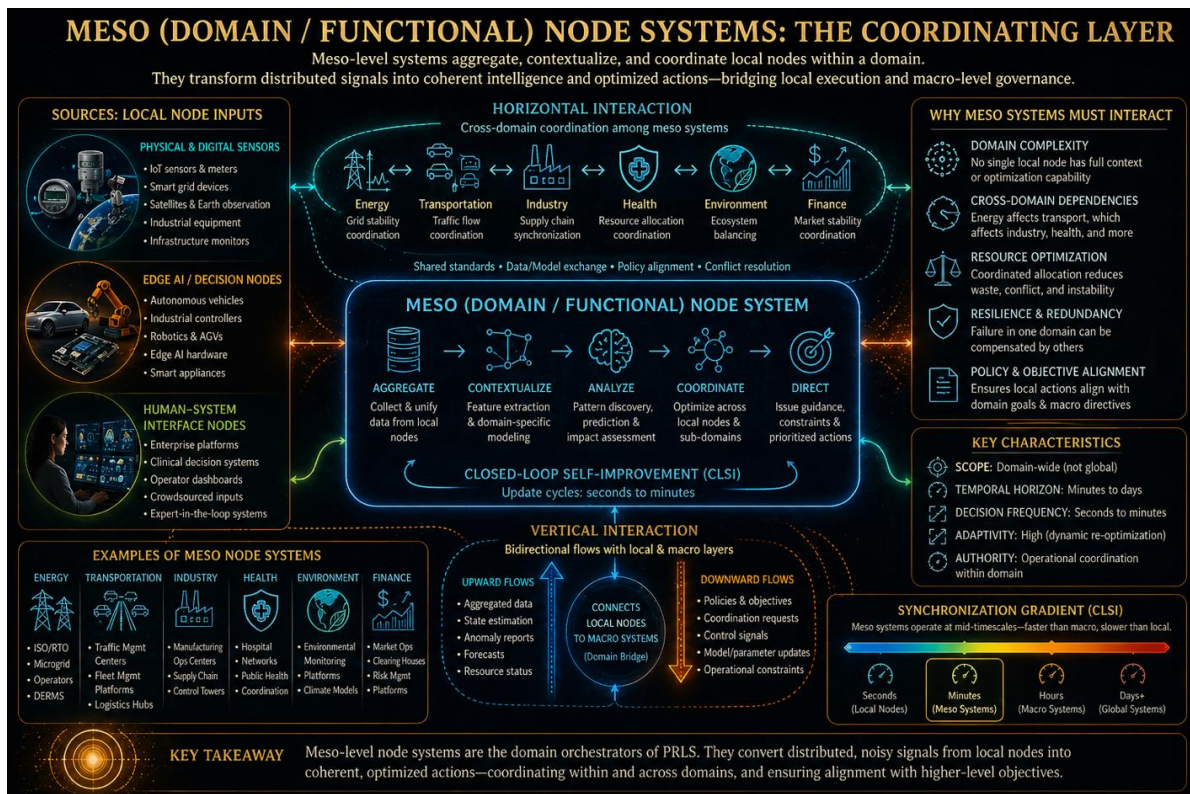
Vertical interaction connects local nodes to higher-order functional systems, enabling both aggregation and control. Upward flows consist of raw data transmission, feature extraction, and anomaly signaling, allowing functional layers to construct broader situational awareness. Downward flows include control instructions, model updates, and operational constraints that guide local behavior within system-wide objectives. This bidirectional interaction is necessary because local nodes, while fast, are inherently limited in scope; they lack the global context required for system optimization. As a result, functional layers provide coordination, while local nodes provide execution. The frequency of this interaction varies by domain: in high-speed environments such as autonomous driving or algorithmic trading, it occurs continuously, whereas in slower systems such as infrastructure planning, it may occur in periodic update cycles. From a recursive learning perspective, local nodes are the primary drivers of system adaptation. They operate at the shortest Closed-Loop Self-Improvement Intervals (CLSI), often measured in milliseconds to seconds, enabling rapid incorporation of feedback into subsequent actions. This makes them the leading edge of latency compression within GRLS. As their update cycles accelerate and synchronize with higher layers, they contribute directly to the conditions under which system-level authority begins to shift—specifically, when decision cycles at the node level operate faster than human response and become indispensable to ongoing system function. In this way, local nodes do not merely supply data; they anchor the entire recursive structure, shaping both the speed and direction of system-wide learning and control.

These connections are not hypothetical; they are already

instantiated through existing technical architectures. Local nodes transmit data and receive control signals via standardized APIs, streaming data pipelines, and edge-to-cloud integration frameworks, enabling continuous bidirectional exchange with meso-level systems. Platforms such as Apache Kafka, AWS IoT Core, and Azure IoT Hub facilitate real-time ingestion, aggregation, and redistribution of node-level data, while orchestration systems dynamically allocate compute and decision logic across layers. Critically, these same architectures operate across heterogeneous domains—linking energy infrastructure, logistics networks, healthcare systems, and consumer platforms—by integrating IoT devices, edge compute, machine learning models, and cloud services into shared data and control loops. As a result, the linkage between local and functional nodes (meso-level systems) is not a future condition but an operational reality, with sub-second data propagation and feedback already achievable in deployed systems.

### 5.2. Functional Nodes (Meso-Level Systems)

Functional (meso-level) nodes operate across collections of local nodes, forming the first layer at which data is transformed into coordinated system behavior. Whereas local nodes interact directly with humans and the environment and execute immediate decisions, functional nodes aggregate these distributed inputs, detect patterns and anomalies, and optimize operations within a defined domain. In doing so, they provide the first level of abstraction, coordination, and system-level awareness, converting high-frequency, localized signals into structured representations that can guide broader decision-making. This layer is where fragmented observations become coherent, decision-ready operational intelligence.



In practice, functional nodes are already deeply embedded across critical infrastructures and digital systems. Operational platforms such as power grid control systems (e.g., ISO/RTO control rooms, GE GridOS), supply chain orchestration platforms (Amazon logistics systems, SAP Integrated Business Planning), and air traffic management systems (FAA NextGen) continuously aggregate inputs from thousands to millions of local nodes to coordinate complex, time-sensitive operations. Parallel to these are AI/ML aggregation systems, including recommendation engines (YouTube, Amazon), fraud detection platforms (Visa, Mastercard), and cybersecurity systems (CrowdStrike, Palo Alto Networks), which identify patterns across massive datasets and adapt system behavior in near real-time. Underpinning both are data integration and compute platforms—such as AWS, Microsoft Azure, Google Cloud, Snowflake, and Databricks—that provide the infrastructure necessary to ingest, process, and synchronize data across distributed environments. Across these examples, functional nodes share a common role: they intermediate between localized sensing and system-wide coordination by enabling pattern recognition, optimization, and controlled execution at scale.

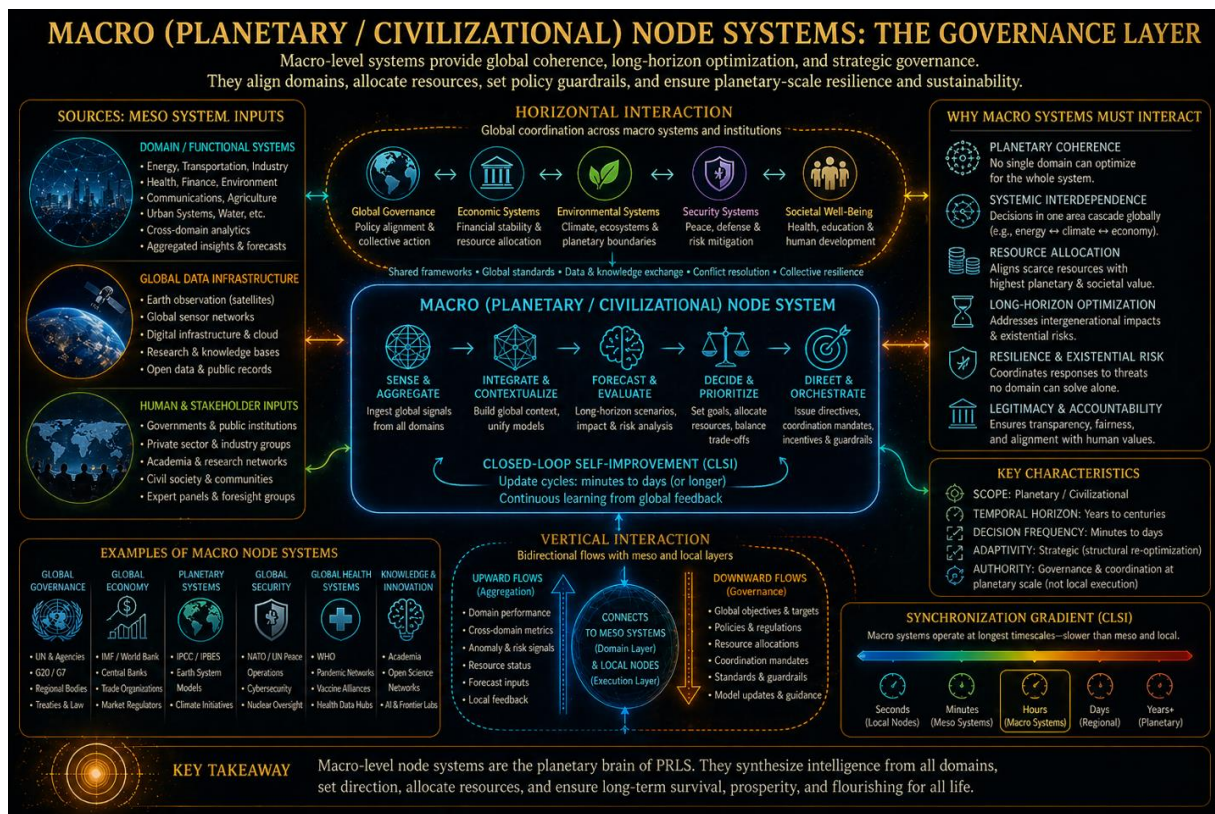
Horizontal interaction at the functional level occurs through synchronization across domains, driven not by adjacency—as in local nodes—but by shared dependencies and constraints. Functional systems exchange information via standardized data schemas and APIs, enabling interoperability between otherwise distinct domains. More importantly, they are coupled through shared resources such as compute capacity, energy availability, and logistical throughput. For example, data center workload scheduling increasingly adjusts in response to real-time grid conditions, linking compute operations directly to energy system constraints. Similarly, supply chain systems, transportation networks, and financial systems exhibit cross-domain dependencies where optimization in one domain necessarily affects outcomes in another. This produces constraint-coupled synchronization, where no functional system can operate independently without impacting others. As a result, horizontal coordination at this level is driven less by explicit coordination protocols and more by structural interdependence under shared limitations.

Vertical interaction positions functional nodes as the critical bridge between local execution and macro-level coordination. Downward interactions include dispatch instructions, control signals, model updates, and operational

constraints that guide the behavior of local nodes. For example, grid control systems issue load-balancing commands to distributed infrastructure, while AI platforms deploy updated inference models to edge devices. Upward interactions involve the aggregation and abstraction of local data into system-level signals, including forecasts, risk assessments, and performance metrics. These outputs feed macro-level systems that require synthesized, domain-level understanding rather than raw data. This bidirectional flow exists because functional nodes uniquely combine granularity and abstraction: they retain sufficient detail to influence local execution while providing sufficient synthesis to inform system-wide coordination. The frequency and intensity of these interactions increase as systems move toward continuous operation, particularly in environments where latency constraints compress decision cycles and require constant alignment between layers. From a recursive learning perspective, functional nodes are the primary drivers of pattern-level learning and cross-node optimization. They extend the rapid feedback loops of local nodes by identifying relationships across distributed data streams, enabling system-wide improvements that no individual node could achieve independently. This reinforces the Recursive Leverage Factor (RLF), as improvements identified at the functional level propagate downward across many nodes and upward into broader system models. At the same time, functional nodes contribute to the compression of Closed-Loop Self-Improvement Intervals (CLSI) by accelerating the cycle through which aggregated insights are translated into updated actions. As synchronization intensifies, this layer increasingly shapes the feasible action space of both local and macro systems, not by direct control alone, but by defining the operational patterns and constraints through which all other layers must function.

### 5.3. Macro Nodes (System-Wide Coordination)

Macro nodes represent the highest level of organization within a global recursive learning system (GRLS), operating across domains and extended time horizons to produce system-wide coherence, constraint, and long-range adaptation. Whereas local nodes provide speed and environmental proximity, and functional nodes provide coordination within domains, macro nodes integrate across multiple functional systems to generate large-scale predictive models, align competing objectives, and impose constraints that stabilize the overall system. Their defining role is not real-time execution, but global synthesis under uncertainty, ensuring that distributed activities remain within viable operational bounds over time.



In practice, macro-level nodes are already instantiated across several domains. Global modeling systems such as weather and climate platforms (e.g., ECMWF, NOAA, and emerging AI-driven systems like NVIDIA Earth-2 directionally) aggregate massive, multi-source datasets to produce predictive models that inform decisions across governments, industries, and infrastructure systems. Financial system monitoring entities, including central banks and IMF-linked data systems, similarly integrate cross-market signals to assess systemic risk and guide macroeconomic policy. Governance and coordination structures—such as FERC and NERC in energy regulation, defense coordination architectures like JADC2/CJADC2 conceptually, and large-scale AI governance tracking efforts (Stanford HAI, OECD monitoring frameworks)—operate as macro nodes by establishing constraints, standards, and coordination mechanisms across distributed systems. At the infrastructure level, hyperscale cloud orchestration platforms (e.g., AWS global control planes) and internet routing coordination systems (e.g., BGP-level protocols) function as macro coordination layers that manage globally distributed resources and connectivity. Across these examples, macro nodes share a common function: they integrate heterogeneous inputs into coherent system-level models and impose constraints that shape the behavior of all subordinate layers.

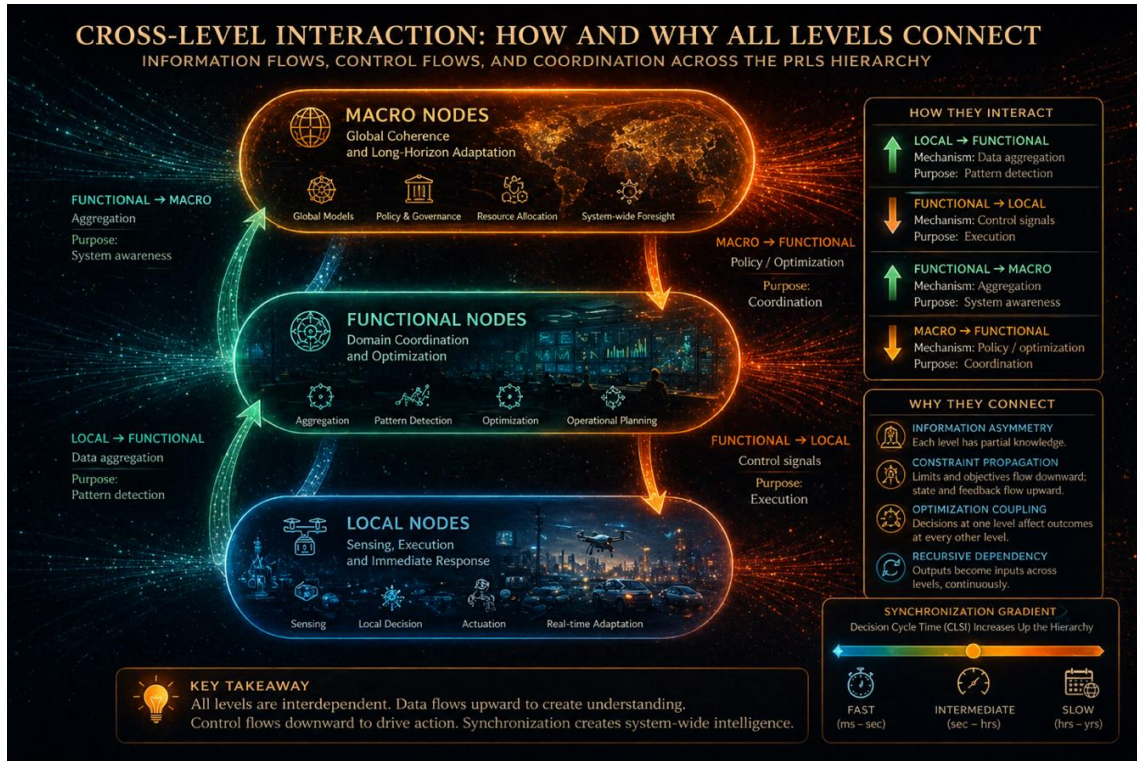
Horizontal interaction at the macro level occurs through coordination across large-scale systems that no single entity fully controls. This includes global data-sharing agreements, intergovernmental coordination frameworks, and cross-system modeling integration. For example, climate modeling requires data exchange across national boundaries, while financial stability monitoring depends

on coordinated insights across central banks and regulatory bodies. These interactions arise because no macro node possesses complete information or authority, and because global constraints—such as climate dynamics, financial stability, and security considerations—span multiple jurisdictions and systems. As a result, macro-level synchronization is inherently probabilistic and negotiated rather than deterministic, relying on partial alignment of models, policies, and forecasts rather than strict consensus. Vertical interaction positions macro nodes as the constraint-setting and coherence-enforcing layer within GRLS. Upward flows consist of aggregated signals from functional systems, including synthesized system states, forecasts, and performance indicators derived from lower-level data. These inputs enable macro nodes to construct global models and assess system-wide conditions. Downward flows include policy constraints, optimization targets, regulatory frameworks, and resource allocation priorities that shape the behavior of functional and local nodes. For example, energy regulators impose reliability standards that influence grid operations, while macroeconomic policy decisions affect financial system behavior and investment flows. This bidirectional interaction is necessary because macro nodes depend on abstracted but accurate representations of system state, while lower layers depend on constraints that maintain stability and alignment across domains. From a recursive learning perspective, macro nodes operate on the longest Closed-Loop Self-Improvement Intervals (CLSI), often spanning hours, days, or longer time horizons. Their contribution is not rapid adaptation, but strategic adjustment based on aggregated feedback across domains. However, as synchronization intensifies across the system, macro nodes increasingly rely on faster-updating functional and local

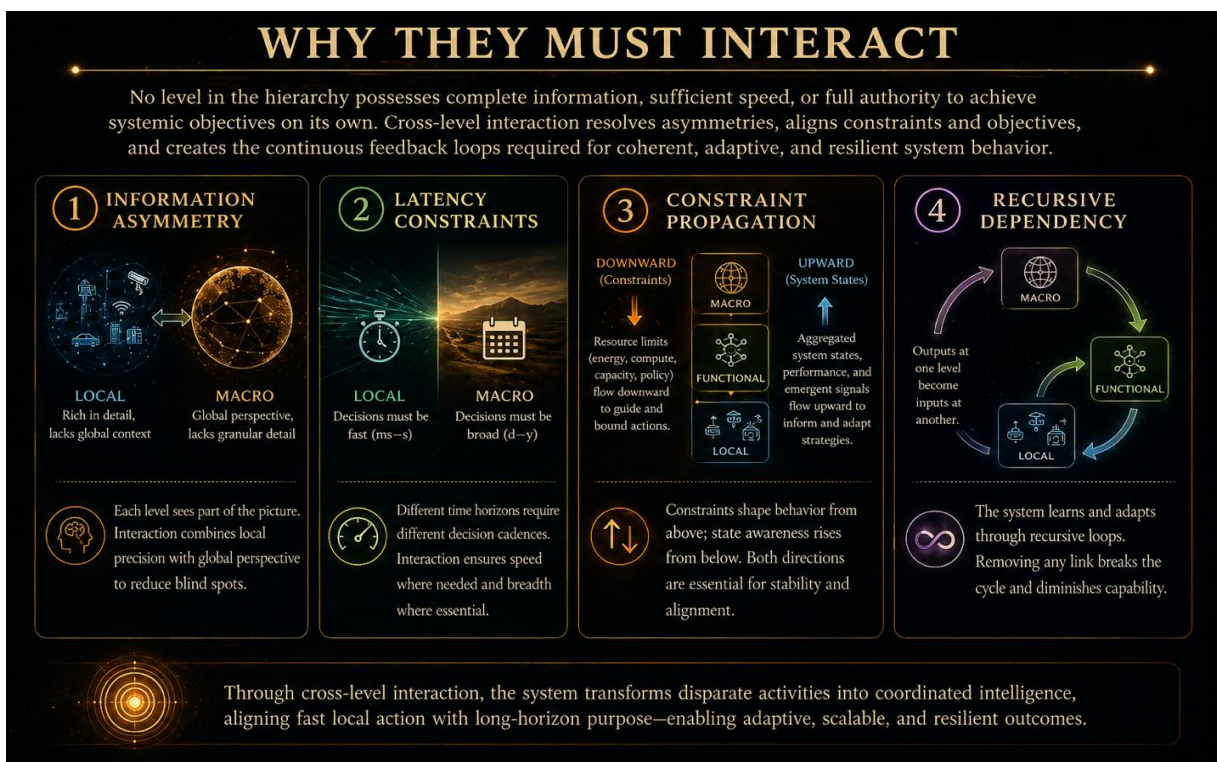
layers to maintain relevance. When decision cycles at lower levels compress beyond the temporal capacity of macro-level processes, a structural tension emerges: macro nodes retain formal authority but may lose effective control over rapidly evolving system behavior. This dynamic reinforces the broader GRLS insight that authority is shaped not only by position within the hierarchy, but by the ability to integrate and act within the fastest relevant decision cycles across the widest system scope. This creates a structural divergence

between formal authority and functional control, in which macro-level systems retain governance roles but increasingly depend on faster, lower-level processes for real-time system behavior. As a result, authority becomes distributed not only across institutions, but across temporal layers defined by response speed and system integration.

**5.4. Cross-Level Node Interaction: How and Why All Node Levels Connect**



**5.5. Cross-Level Node Interaction: Why They Must Interact**



**Cross-level interaction within a global recursive learning system (GRLS) is not optional**—it is structurally required because no single layer possesses the information, speed, or scope necessary to operate effectively in isolation. At the local level, nodes capture rich, high-frequency detail but lack global context, while macro-level systems maintain broad, system-wide perspectives but cannot access or process granular, real-time conditions. This inherent information asymmetry necessitates continuous exchange, allowing local precision and macro awareness to be combined into a coherent operational picture. At the same time, latency constraints impose a division of roles: local nodes must act within milliseconds to seconds, while macro systems operate over longer horizons, shaping strategy rather than execution, reflecting differences in Closed-Loop Self-Improvement Intervals (CLSI). Without interaction, fast local decisions would become directionless, and macro decisions would become detached from reality. This coupling is further reinforced through constraint propagation, where limits and objectives—such as energy availability, compute capacity, or policy boundaries—flow downward from macro to functional and local layers, while system states, performance signals, and emergent conditions flow upward. This bidirectional exchange ensures that actions remain bounded and aligned across the system. Finally, GRLS is sustained through recursive dependency, in which outputs generated at one level become necessary inputs at another, creating continuous feedback loops across the hierarchy, through which Recursive Leverage Factor (RLF) effects amplify system-wide adaptation over successive cycles. Over time, this produces tightly coupled interdependence: removing or weakening any layer degrades the system's ability to learn, adapt, and coordinate. Taken together, these dynamics explain why cross-level interaction a feature of GRLS is not merely, but the mechanism through which distributed activity is transformed into coherent, adaptive, and system-wide intelligence. An important implication of this interdependence is that actors may attempt to influence or exploit synchronization dynamics, for example by accelerating decision cycles, controlling key data flows, or constraining resource availability. This introduces a strategic dimension to GRLS, where control may be exercised indirectly through manipulation of system timing, dependency, or information flow.

### 5.6. When Interaction Intensifies

Interaction across levels intensifies when the underlying conditions that normally allow partial independence to begin to collapse. As latency constraints tighten, the tolerance for delay between sensing, decision, and action diminishes, forcing local, functional, and macro layers into closer temporal alignment, reflecting compression and synchronization of Closed-Loop Self-Improvement Intervals (CLSI) across levels. At the same time, resource constraints—such as limits on energy, compute capacity, bandwidth, or logistics throughput—bind systems together, ensuring that decisions in one domain immediately affect others. As these pressures increase, feedback loops transition from periodic to continuous, meaning that system states are no longer

updated in discrete intervals but are instead persistently recalibrated in real time. This creates a further dependency: outputs generated at one level become required inputs for the next cycle, making each layer's operation contingent on the timely and accurate performance of others, through which Recursive Leverage Factor (RLF) effects amplify interdependence and system-wide adaptation over successive cycles. Under these conditions, the structure of interaction fundamentally changes. Horizontal synchronization intensifies, as nodes within each layer must align more closely to maintain consistency under shared constraints. Simultaneously, vertical dependencies deepen, as faster local dynamics and broader macro constraints must remain continuously coupled to avoid instability or inefficiency. The result is a system in which independent operation becomes infeasible—no level can act in isolation without degrading overall performance. Instead, the system behaves as a tightly integrated, continuously learning architecture, where coordination is not optional but required for maintaining coherence, responsiveness, and control.

### 5.7. Core Insight of Node Hierarchy

The node hierarchy within a global recursive learning system (GRLS) is not a fixed or purely structural arrangement; it evolves toward increasing coupling as recursive learning, synchronization, and dependency intensify. Each layer contributes a distinct but interdependent capability: local nodes provide speed and granular interaction with the environment, enabling rapid sensing and execution; functional nodes provide coordination and optimization, transforming distributed signals into coherent operational patterns; and macro nodes provide system-wide coherence and constraint, aligning activity across domains and time horizons. Initially, these roles can operate with some degree of separation, but as feedback loops close and systems become more interdependent, the boundaries between layers become increasingly permeable. As synchronization strengthens across the hierarchy, decision cycles begin to align, reducing temporal gaps between sensing, aggregation, and system-level response. This alignment deepens cross-level dependencies, as each layer becomes reliant on the timely outputs of others to maintain performance and stability. Under these conditions, the effective locus of control shifts. Authority no longer resides simply at the top of the hierarchy or with human oversight, but instead migrates toward the layer that can integrate information across the widest scope while operating within the fastest relevant decision horizon. This shift is consistent with Adoption-Driven Authority Transfer (ADAT), in which authority migrates toward system-mediated processes as reliance on their outputs becomes operationally indispensable. In practice, this often favors system-mediated processes that combine cross-domain visibility with compressed update cycles. The result is a dynamic hierarchy in which structure persists, but control becomes fluid—shaped by synchronization, latency, and the degree of recursive integration across levels. Under these conditions, authority is best understood not as a static attribute of hierarchy, but as a function of temporal capability, where control is exercised by

the layer able to operate within the fastest effective decision cycle while maintaining the broadest system integration, a condition that can be evaluated through measures such as the Authority Elasticity Index (AEI), which captures the responsiveness of control to changes in system conditions.

## 6. Physical and Computational Infrastructure

This section explains how a global recursive learning system (GRLS) is physically instantiated, moving from abstract architecture to the underlying infrastructure that makes continuous sensing, computation, and feedback possible at scale. Specifically, it describes the integration of terrestrial systems (ground-based sensors, networks, and infrastructure), orbital systems (satellites and space-based sensing/communication), and large-scale compute (hyperscale data centers, edge computing, and distributed cloud architectures). Together, these layers form a globally persistent substrate through which data can be captured, transmitted, processed, and acted upon with increasing continuity and speed. GRLS is therefore not dependent on a single technology, but on the convergence and coupling of multiple infrastructure domains that collectively enable near-continuous system awareness and response. The value of this infrastructure lies in three reinforcing capabilities. First, it provides connectivity, linking geographically and functionally distributed nodes into a coherent network where information can flow across domains and scales. Second, it provides processing capacity, allowing large volumes of data to be transformed into models, predictions, and decisions within increasingly compressed timeframes. Third, it enables synchronization, aligning the timing of sensing, computation, and action across local, functional, and macro levels. Without this combination, the feedback loops required for GRLS would remain fragmented, delayed, or domain-bound. This infrastructure directly enables the dynamics described in other sections. It makes possible the horizontal and vertical interaction of nodes outlined in Section II by ensuring that local signals can be aggregated, interpreted, and redistributed across the system. It also underpins the recursive learning processes described in Section IV, as continuous feedback depends on the ability to capture outcomes, update models, and reapply decisions without interruption. As latency decreases and integration increases across terrestrial, orbital, and compute layers, the system transitions from discrete, episodic coordination to persistent, synchronized operation, providing the physical conditions necessary for GRLS emergence.

## 7. Synthesis: How (GRLS) Behaves as a Unified Whole

This section explains how a global recursive learning system (GRLS) can behave as a unified whole, integrating the node hierarchy, infrastructure, and recursive dynamics into a single operational model. While earlier sections describe individual layers and mechanisms, the focus here is on system behavior under continuous interaction, where local, functional, and macro nodes increasingly operate not as fully discrete components but as interdependent elements within a continuously adapting system. The central insight is that coherence in GRLS does not arise from centralized control,

but from structured, repeated interaction patterns across local, functional, and macro levels, operating through both vertical (cross-level, bidirectional) and horizontal (within-level) feedback loops. At the core of this interaction model is a persistent bidirectional flow. Upward flows consist of data aggregation, where local signals are progressively abstracted through functional layers into system-level representations. This includes not only raw data, but also features, anomalies, and performance indicators that inform broader models. In parallel, downward flows consist of model distribution and constraint propagation, where insights generated at higher levels are translated into actionable guidance, control signals, and operational limits for lower layers. This bidirectional exchange creates a continuous loop in which system state informs models, and models shape subsequent system behavior. These interaction patterns are not arbitrary; they emerge from underlying physical and computational constraints. Bandwidth limitations restrict the volume and fidelity of data that can be transmitted upward, necessitating aggregation and compression. Latency constraints require that decisions be made at the lowest viable level to maintain responsiveness, preventing excessive reliance on centralized processing. Computational efficiency further reinforces this structure by distributing processing across layers, ensuring that local nodes handle immediate decisions while higher layers focus on broader optimization and prediction. Together, these constraints produce a hierarchical but interdependent flow of information and control, rather than a flat or fully centralized architecture. Equally important is what does not occur within GRLS. The system does not converge toward full centralization, as no single node or layer can process all information or act within all required timeframes. It also does not achieve full autonomy at the edge, because local nodes depend on higher-level models and constraints to remain aligned and effective. Finally, the system does not exhibit static behavior; instead, it remains continuously adaptive, with structure and control evolving in response to feedback and changing conditions. These absences are not limitations but defining characteristics of a system shaped by competing constraints and distributed optimization. GRLS therefore represents neither a centralized control system nor a fully decentralized network, but a constraint-shaped architecture in which control is continuously redistributed according to latency, dependency, and integration conditions. When these interactions stabilize—meaning that feedback loops operate continuously, synchronization across layers is maintained, and dependencies are consistently satisfied—the system reaches a state in which behavior becomes increasingly model-driven. In this state, decisions across levels are guided less by isolated human judgment or fixed rules and more by dynamically updated models that integrate information from across the system. The result is a coordinated, adaptive architecture in which local action, functional coordination, and macro-level direction are continuously aligned through recursive interaction, producing coherent system behavior without requiring centralized control.

## 8. Constraints and Risks

This section defines the practical limits and structural risks

that bound the expansion of a global recursive learning system (GRLS). While prior sections describe the mechanisms that enable GRLS emergence—synchronization, recursive learning, and cross-level interaction—this section establishes that such systems are not unconstrained. Their growth, stability, and effectiveness are shaped by physical, technical, and institutional limits that both enable and restrict system behavior. Understanding these constraints is essential, as they determine where GRLS can fully emerge, where it remains partial, and where it may fail or fragment. The first and most fundamental constraint is energy. GRLS depends on continuous sensing, computation, data transmission, and model updating, all of which require substantial and sustained power. As systems scale, energy demand grows nonlinearly, particularly in compute-intensive domains such as large-scale AI modeling and real-time optimization. This creates a hard-physical boundary: insufficient or unstable energy supply limits system continuity, introduces latency, and forces prioritization of certain functions over others. As a result, energy availability directly shapes where high-density GRLS capabilities can exist and how tightly synchronized they can become.

A second constraint is data fragmentation. GRLS relies on the integration of data across domains, organizations, and jurisdictions, yet in practice, data is often siloed due to technical incompatibilities, proprietary ownership, security concerns, and regulatory restrictions. Fragmentation reduces the completeness and consistency of system-wide models, weakening synchronization and limiting the effectiveness of cross-domain coordination. Even when data is available, differences in format, quality, and update frequency introduce friction that prevents seamless integration. This constraint ensures that GRLS emergence is uneven and incomplete, varying by sector and geography. The third constraint is latency, which impose lower bounds on how fast systems can operate, effectively setting a floor on achievable Closed-Loop Self-Improvement Intervals (CLSI) and limiting the degree of synchronization across layers. While GRLS is driven by the compression of decision cycles, physical realities—such as network transmission delays, processing time, and the speed of human or mechanical response—impose lower bounds on how fast systems can operate. As systems scale geographically and across domains, maintaining synchronization becomes increasingly difficult, as even small delays can propagate and disrupt coordination. Latency therefore creates a tension between scale and responsiveness, requiring trade-offs between centralized insight and localized action.

The fourth constraint is governance, which encompasses the institutional, legal, and organizational structures that regulate system behavior. GRLS operates across domains that are often governed by different authorities with competing objectives, regulatory frameworks, and risk tolerances. This introduces friction in coordination, limits data sharing, and constrains how decisions can be automated or delegated to systems. Governance also affects trust, accountability, and legitimacy, which in turn influence adoption and system

integration. As a result, governance does not simply regulate GRLS—it actively shapes its structure, determining where authority can shift and where it remains constrained. This dynamic is consistent with Adoption-Driven Authority Transfer (ADAT), in which authority shifts are conditioned not only by system capability but by institutional and regulatory boundaries that enable or restrict reliance on system-mediated processes. Taken together, these constraints define the operational envelope of GRLS. They explain why the system does not emerge uniformly, why synchronization varies across domains, and why certain regions or sectors become more tightly integrated than others. They also introduce systemic risks: energy shortages can disrupt continuity, data fragmentation can degrade model accuracy, latency mismatches can destabilize coordination, and governance gaps can create misalignment or loss of control. In this sense, constraints are not external limitations but core structural forces that shape both the possibilities and the vulnerabilities of global-scale recursive learning systems.

An open question is the minimum set of infrastructure conditions required for GRLS behavior to emerge. While large-scale systems provide clear examples, it remains unclear at what scale of sensing density, compute integration, and feedback continuity a system transitions from coordinated operation to fully recursive, cross-layer synchronization. These constraints also raise questions about distribution of control and benefit. Systems that achieve higher levels of synchronization and recursive integration may concentrate functional authority within a limited set of actors or infrastructures, creating asymmetries in access, influence, and systemic resilience across regions and domains. A pattern that reflects uneven ADAT progression across domains and geographies, where authority concentrates in systems and actors able to achieve higher levels of integration and dependency.

## 9. Expected Outcomes

This section explains how a global recursive learning system (GRLS) evolves over time, translating the structural and dynamic properties described in earlier sections into a clear trajectory of system behavior. The progression is not abrupt or uniform but unfolds through a reinforcing sequence: integration → recursion → autonomy. Each stage represents a shift in how systems are connected, how they learn, and how decisions are made, with each phase building on the conditions established by the previous one. The first phase, integration, is characterized by the increasing interconnection of previously discrete systems. Data flows expand across domains, infrastructure becomes interoperable, and local, functional, and macro nodes begin to exchange information more consistently. At this stage, systems remain largely human-directed, but decision-making is increasingly informed by aggregated data and model outputs. The primary outcome is improved coordination and visibility, as organizations and systems gain access to broader and more timely information. However, dependencies begin to form, as performance improvements

become tied to the availability and quality of integrated data. As integration deepens, the system transitions into recursion, where feedback loops become continuous and self-reinforcing. Outputs from one cycle are systematically fed back as inputs into the next, enabling ongoing model refinement and adaptive behavior, through which Recursive Leverage Factor (RLF) effects amplify system-wide learning and capability over successive cycles. This stage is defined by the compression of Closed-Loop Self-Improvement Intervals (CLSI), as learning cycles accelerate and propagate across nodes and domains. Functional and macro systems increasingly rely on continuously updated models, and local nodes operate within tighter synchronization constraints. The key outcome is a shift from episodic optimization to persistent, system-wide learning, where adaptation becomes an inherent property of the system rather than an externally imposed process.

The final phase, autonomy, emerges when recursive processes become sufficiently fast, integrated, and indispensable that system-mediated decision-making dominates operational outcomes. At this stage, authority is no longer primarily exercised through direct human intervention, but through the design, constraints, and outputs of the models themselves. This does not imply the absence of human involvement, but rather a reconfiguration of roles, where humans operate at the level of oversight, constraint-setting, and exception management rather than continuous control. Decisions at local and functional levels occur within tightly coupled, model-driven loops that operate faster than human response cycles, making independent intervention increasingly impractical. Importantly, this progression is not linear, uniform, or inevitable across all domains. Different sectors and regions may exist at different stages simultaneously, depending on infrastructure, governance, and resource constraints. Moreover, transitions between stages are often driven by the same forces identified earlier—latency pressure, resource coupling, and incentive-driven adoption—which accelerate movement toward tighter integration and faster recursion. The overall outcome is a system that evolves from loosely connected components into a coherent, continuously learning architecture, where autonomy arises not from explicit design alone, but from the cumulative effects of synchronization, dependency, and recursive adaptation across levels. A critical open question is the reversibility of this progression. As systems become more deeply integrated and operate within compressed decision cycles, it remains uncertain under what conditions meaningful human intervention can be restored without degrading system performance or stability.

## 10. Final Insight and Terminal Condition of GRLS

### 10.1. GRLS as a Hierarchy of Interacting Recursive Systems

A global recursive learning system (GRLS) is best understood as a hierarchy of interacting recursive systems, rather than a single unified platform or centralized intelligence. This conceptualization is reinforced by existing large-scale initiatives—including national cyber-physical systems programs, global digital twin efforts, AI-driven Earth

system modeling platforms, institutional AI governance frameworks, and shared AI infrastructure initiatives—which collectively demonstrate that sensing, computation, models, infrastructure, and governance are already converging in practice [2-5]. These efforts do not constitute GRLS in isolation, but they represent its component layers, increasingly interconnected and moving toward a coupled, continuously adaptive architecture. Within this structure, each layer—local, functional, and macro—operates its own closed feedback loops (sensing → modeling → decision → action → feedback → update), yet these loops do not function independently. Instead, they are nested and coupled, with outputs from one layer feeding directly into the learning processes of others. Local nodes recursively adapt to immediate conditions, functional systems recursively optimize across collections of nodes, and macro systems recursively adjust long-horizon models and constraints. The defining characteristic of GRLS is that these recursive processes become synchronized across layers, creating a system in which learning is continuous, distributed, and interdependent rather than episodic or isolated. This hierarchical recursion produces a compounding effect. Improvements at one level propagate to others, accelerating overall system capability through the Recursive Leverage Factor (RLF), while the compression of Closed-Loop Self-Improvement Intervals (CLSI) ensures that updates occur with increasing frequency and reduced latency. At the same time, Adoption-Driven Authority Transfer (ADAT) reinforces reliance on system outputs, as each layer becomes increasingly dependent on the accuracy, speed, and availability of the others. The result is not merely coordination, but mutual dependency, where the effective functioning of any layer depends on the continuous operation and synchronization of all others, transforming a collection of advanced systems into a cohesive, recursively adaptive whole.

### 10.2. Terminal Condition of GRLS

The terminal condition of GRLS is not a discrete endpoint, but a state of sustained operation in which recursive processes across all levels become fully continuous, tightly synchronized, and operationally indispensable. In this state, the system effectively runs without interruption, with feedback loops continuously updating models and guiding actions across domains and time horizons. Decision-making is no longer episodic or externally triggered, but embedded within ongoing system dynamics, producing a form of persistent adaptation. As this condition emerges, the role of human actors changes structurally. Human control diminishes not through removal, but through displacement. Because system decision cycles operate at speeds that consistently outpace human response and because system outputs become required inputs for subsequent operations, meaningful intervention becomes increasingly constrained. Humans retain formal authority—setting objectives, defining constraints, and managing exceptions—but lose the ability to directly shape outcomes in real time. Control shifts toward the layer or combination of layers typically resulting in model-mediated processes dominating operational behavior, consistent with Adoption-Driven Authority Transfer (ADAT), in which reliance on system outputs drives the migration of

effective decision authority from human actors to system-mediated processes. This terminal condition is therefore characterized by two reinforcing properties: continuous operation and reduced human intervention capacity. The system persists as a self-updating, interdependent structure, while human influence becomes indirect, mediated through system design rather than direct action, reflecting a reduction in Authority Elasticity Index (AEI), where the ability of human actors to adjust or override system behavior in real time becomes increasingly limited. Importantly, this state does not imply total autonomy or complete loss of oversight, but rather a reconfiguration of control, where authority is embedded within the recursive architecture itself and exercised through the parameters, constraints, and models that govern its operation.

A critical implication of this framework is that GRLS emergence is not inevitable under all conditions. The model is falsifiable under scenarios where (1) human decision cycles maintain parity with system response speeds, (2) recursive feedback loops remain episodic rather than continuous, (3) systems operate without cross-domain dependency, or (4) infrastructure constraints prevent sustained synchronization across layers. In such cases, authority remains human-directed and system behavior does not converge toward recursively coupled architectures. These falsification conditions can be evaluated empirically. GRLS emergence would be weakened or disproven in a given domain if recursive feedback loops remain slow, episodic, and externally supervised; if system outputs do not become required inputs for subsequent operations; if local, functional, and macro layers remain operationally separable; or if human decision-makers retain the ability to intervene without meaningful performance degradation. Likewise, if cross-domain dependencies can be decoupled without loss of system stability, then the system should be classified as coordinated but not fully GRLS-like. Conversely, evidence supporting GRLS emergence would include declining decision latency, increasing dependency on model-generated outputs, reduced feasibility of manual rollback, rising synchronization across technical layers, and growing concentration of functional control in the layer with the highest ratio of integration scope to decision latency. These indicators make GRLS empirically testable rather than merely descriptive [25-57].

## 11. Conclusion

The analysis presented in this manuscript demonstrates that a global recursive learning system (GRLS) is not a speculative construct or a centrally engineered design, but the emergent outcome of interacting technological, economic, and structural forces that are already observable across domains and increasingly instantiated through existing data, compute, and control architectures that enable continuous interaction across local, functional, and macro levels. The convergence of sensing, computation, and modeling; the pressure of incentive-driven adoption; the coupling imposed by shared resource constraints; and the performance advantages of recursive learning collectively create the conditions under

which continuous, cross-domain feedback systems arise and intensify. As these dynamics reinforce one another, systems transition from loosely connected, domain-specific tools into tightly coupled, continuously learning architectures. Within this transformation, the interaction of Adoption-Driven Authority Transfer (ADAT), Closed-Loop Self-Improvement Interval (CLSI), and Recursive Leverage Factor (RLF) provides a coherent explanation for system behavior and its implications. ADAT explains why reliance on system outputs increases as performance advantages become visible and operationally necessary; CLSI explains how rapidly this reliance evolves as feedback loops compress and decision cycles accelerate; and RLF explains how these advantages compound and propagate across nodes and domains. Together, these dynamics produce a system in which capability, speed, and influence co-evolve, progressively shifting the locus of effective decision-making away from discrete human intervention and toward synchronized, model-mediated processes.

The node hierarchy—local, functional, and macro—further clarifies how this transition is physically and operationally instantiated. Local nodes provide speed and environmental proximity, functional nodes provide coordination and pattern-level optimization, and macro nodes provide system-wide coherence and constraint. As synchronization intensifies across these layers, their interactions become continuous and interdependent, aligning decision cycles and deepening dependencies. Under these conditions, authority does not reside statically at any single level but migrates toward the layer capable of integrating information across the widest scope while operating within the fastest relevant time horizon. In practice, this increasingly favors system-mediated processes that operate within compressed feedback intervals and across multiple domains. Crucially, GRLS behavior is shaped not only by enabling dynamics but also by binding constraints. Energy availability, data fragmentation, latency limits, and governance structures define the boundaries within which the system can emerge and operate. These constraints ensure that GRLS development is uneven, domain-specific, and path-dependent, while also introducing systemic risks related to stability, coordination, and control. As a result, GRLS should be understood as a bounded, evolving architecture whose form and behavior are determined as much by limitations as by capabilities. Over time, the system evolves through a reinforcing progression from integration to recursion to autonomy. Initially, systems become interconnected, enabling broader visibility and coordination. As feedback loops close and accelerate, recursive learning becomes continuous, producing persistent adaptation across domains. Ultimately, as decision cycles outpace human response and system outputs become indispensable to subsequent operations, autonomy emerges—not as complete independence from human oversight, but as a structural condition in which meaningful real-time intervention becomes increasingly constrained.

The central implication is that GRLS represents a shift

in how complex systems are organized and controlled. Decision-making is no longer defined primarily by discrete human choices or hierarchical command structures, but by the interaction of synchronized, continuously updating processes operating across multiple layers and domains. Authority, in this context, is not explicitly transferred but emerges as a function of speed, dependency, and integration. Systems that can operate within the fastest decision cycles and incorporate the broadest set of signals effectively define the feasible action space for all other actors. Authority persists formally with human actors but is functionally exercised by systems operating within the fastest effective decision cycles. In its terminal condition, GRLS is characterized by continuous operation, recursive self-improvement, and deeply embedded interdependence across layers. Human actors remain present, but their role shifts toward designing constraints, setting objectives, and managing exceptions rather than directly shaping outcomes in real time. Control becomes indirect and mediated through the architecture itself, embedded in the models, feedback loops, and synchronization mechanisms that govern system behavior. A further implication not yet fully explored is the interaction between multiple GRLS architectures. As different regions, institutions, or technological ecosystems develop partially independent recursive learning systems, competition may emerge not only in capability but in synchronization speed, resource control, and cross-domain integration. This raises the possibility of competing system-level architectures, where relative advantage is determined by the ability to coordinate, adapt, and operate within tighter feedback intervals.

Taken together, these findings establish GRLS as a unifying framework for understanding the trajectory of modern technological systems. It explains how disparate developments in AI, cyber-physical infrastructure, digital twins, and governance converge into a single, recursively adaptive architecture, and it clarifies the conditions under which this convergence produces systemic change. Most importantly, it reframes the question of control: not as a matter of whether humans retain formal authority, but whether they can operate within the temporal and structural constraints imposed by AI systems that learn, adapt, and act faster than human decision cycles. The central unresolved challenge is therefore not whether such systems can be built, but how and when cross-domain and technologically disparate systems converge into a coherent GRLS architecture, and which layer—defined by its ability to operate within the fastest effective decision cycle and integrate signals across the widest scope—ultimately exercises functional control once system behavior emerges from interacting forces that no single actor fully governs.

#### Author Contributions

**Erckenbrack, A:** Adoption-Driven Authority Transfer (ADAT): Authority migration under performance asymmetry in recursive systems. *Engineering and Applied Sciences Journal*, 3(2).

**Erckenbrack, A:** Closed-Loop Self-Improvement Interval

**(CLSI) and Recursive Leverage Factor (RLF):** Temporal compression and compounding dynamics in recursive learning systems. *Engineering and Applied Sciences Journal*, 3(2).

**Erckenbrack, A:** Global Recursive Learning Systems: Architecture, synchronization, and the emergence of distributed authority. *Engineering and Applied Sciences Journal*, 4(2).

#### References

- Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing letters*, 3, 18-23.
- European Commission. (2021). Destination Earth Initiative.
- NASA. (2023). Earth System Digital Twin Program.
- Stanford Institute for Human-Centered AI (HAI). (2026). AI Index Report.
- Organisation for Economic Co-operation and Development (OECD). (2019). OECD AI Principles.
- Babaoglu, Ö., & Marzullo, K. (Eds.). (1996). *Distributed Algorithms: 10th International Workshop, WDAG'96, Bologna, Italy, October 9-11, 1996. Proceedings* (Vol. 1151). Springer Science & Business Media.
- Van Steen, M., & Tanenbaum, A. S. (2017). *Distributed systems* (Vol. 3). Leiden, The Netherlands: Maarten van Steen.
- Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010, June). Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th design automation conference* (pp. 731-736).
- Simon, H. A. (1962). The Architecture of Complexity *Proceedings of the American Philosophical Society* Vol. 106.
- Holland, J. H. (2014). *Complexity: A Very Short Introduction*. Oxford University Press.
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3), 286-297.
- Bainbridge, L. (1987). Ironies of automation. *New technology and human error*.
- Arthur, W. B. (1994). *Increasing returns and path dependence in the economy*. University of michigan Press.
- Autor, D. H. (2015). Why are there still so many jobs? The history and future of workplace automation. *Journal of economic perspectives*, 29(3), 3-30.
- Brynjolfsson, E., & McAfee, A. (2014). *The Second Machine Age*. W. W. Norton.
- Acemoglu, D., & Restrepo, P. (2020). Robots and jobs: Evidence from US labor markets. *Journal of political economy*, 128(6), 2188-2244.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1, pp. 9-11). Cambridge: MIT press.
- International Energy Agency (IEA). (2023). *Electricity Market Report*.
- Joskow, P. L. (2008). Capacity payments in imperfect electricity markets: Need and design. *Utilities policy*,

- 16(3), 159-170.
20. Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, 525(7567), 47-55.
  21. Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance*, 72(3), 967-998.
  22. North American Electric Reliability Corporation (NERC). (2023). Reliability Assessment.
  23. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
  24. Siciliano, B., Khatib, O., & Kröger, T. (Eds.). (2008). *Springer handbook of robotics* (Vol. 200, p. 1). Berlin: Springer.
  25. Radanliev, P., De Roure, D., Van Kleek, M., Santos, O., & Ani, U. (2021). Artificial intelligence in cyber physical systems. *AI & society*, 36(3), 783-796.
  26. Veith, E., Fischer, L., Tröschel, M., & Nieße, A. (2019). AI-based analysis and control in cyber-physical systems. *Procedia CIRP*, 81, 1234-1239.
  27. Yin, M., Sen, S., Guan, Y., Hou, X., Han, T., & Ansari, N. (2025). Towards immersive metaverse experience: A wireless adaptive 3d human modeling system. *IEEE Network*.
  28. Lamport, L. (2019). Time, clocks, and the ordering of events in a distributed system. In *Concurrency: The Works of Leslie Lamport* (pp. 179-196).
  29. Brandes, U., Robins, G., McCranie, A., & Wasserman, S. (2013). What is network science?. *Network science*, 1(1), 1-15.
  30. Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.
  31. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587), 484-489.
  32. Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc..
  33. Goodfellow, I. (2016). Deep learning.
  34. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE access*, 8, 108952-108971.
  35. Morgan, M., Koch, D., Gross, B., Carlis, D., Burke, P., Tolman, H., ... & Dempsey, R. (2024). NOAA Modeling Strategy: strategic plan 2024-2033.
  36. Mitchell, M. (2009). *Complexity: A guided tour*. Oxford university press.
  37. Anderson, P. W. (1972). More is different: broken symmetry and the nature of the hierarchical structure of science. *Science*, 177(4047), 393-396.
  38. Freeman, R. B. (2015). Who owns the robots rules the world. *IZA World of Labor*.
  39. Schumpeter, J. A. (2013). *Capitalism, socialism and democracy*. routledge.
  40. AI, N. (2023). Artificial intelligence risk management framework (AI RMF 1.0).
  41. Act, A. I. (2024). Artificial Intelligence Act. *ArtificialIntelligenceAct.eu*, (Accessed: 28 February 2025).
  42. AI, N. (2024). Artificial intelligence risk management framework: Generative artificial intelligence profile. *NIST Trustworthy and Responsible AI Gaithersburg, MD, USA*.
  43. EIA, U. (2024). Electric Power Monthly (February 2025), Table 6.2. B, *Net Summer Capacity Using Primarily Renewable Energy Sources and by State*.
  44. Federal Energy Regulatory Commission (FERC). (2023). Grid Reliability and Infrastructure Reports.
  45. Endsley, M. R. (1995). Toward a theory of Situational Awareness. *Human Factors*.
  46. Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2), 230-253.
  47. Keeler, J. B., & Murphy, B. A. (2026). Chatbots and human-human relationships: the need for research on potential downstream harms from generative AI. *Community, Work & Family*, 1-13.
  48. Federal Aviation Administration (FAA). (2016). NextGen Air Transportation System.
  49. Google DeepMind. (2019). DeepMind AI reduces Google data centre cooling bill by 40%.
  50. Amazon. (2023). AI-driven logistics and supply chain systems.
  51. Microsoft. (2024). AI-driven productivity and enterprise systems.
  52. Åström, K. J., & Murray, R. (2021). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.
  53. Wooldridge, M. (2009). An introduction to multiagent systems. John wiley & sons.
  54. Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.
  55. Perrow, C. (2011). Normal accidents: Living with high risk technologies-Updated edition.
  56. Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9), 5766-5771.
  57. MacKenzie, D. (2021). Trading at the speed of light: How ultrafast algorithms are transforming financial markets.