

GOVERNANCE-AS-CODE FOR REAL-TIME AI SYSTEMS: CROSS-LAYER APPROACH

Davronbekov Dilmurod Abdujalilovich

DSc, Professor, Department of Mobile Communication Technologies, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.

Saydirasulov Norkobil Saydirasulovich

PhD, DSc Student, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan.

Khidirov Abduvali Makhmadaliyevich

Associate Professor, Department of Telecommunication Engineering and Infocommunication Systems, Samarkand Branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Samarkand, Uzbekistan.

Abstract. Real-time artificial intelligence (AI) systems in cyber-physical and Internet-of-Things (IoT) environments face critical challenges due to fragmentation across sensing, computing, and networking layers. While advances in AIoT, edge intelligence, and distributed learning improve individual components, they lack coordinated system-level control. This paper introduces governance-as-code as a unifying paradigm that embeds executable policies directly into runtime, enabling continuous monitoring, adaptive decision-making, and cross-layer coordination. A unified taxonomy and policy-driven framework are proposed to improve reliability, scalability, and adaptability of real-time AI systems.

Keywords: Governance-as-Code, Real-Time AI, AIoT, Edge Intelligence, Cross-Layer Systems

1. Introduction

Real-time artificial intelligence (AI) systems are increasingly deployed in cyber-physical and Internet-of-Things (IoT) environments, where sensing, computation, and communication must operate under strict latency, reliability, and resource constraints. Applications such as autonomous transportation, smart grids, and industrial automation require continuous coordination across distributed components in dynamic environments. Recent advances in AIoT and edge intelligence enable scalable AI services [1–6], while distributed learning improves decentralized intelligence [7–9].

However, existing systems remain fragmented across sensing, computing, and networking layers, limiting system-level optimization. Increasing sensing quality improves accuracy but increases computational and communication costs, while dynamic network conditions affect inference placement and responsiveness [10–13].

In addition, traditional AI governance approaches rely on static mechanisms such as auditing and compliance [14–16], which are insufficient for real-time adaptive systems. To address these challenges, this work introduces governance-as-code, embedding policies directly into system runtime for adaptive and coordinated control.

2. Cross-Layer Architecture

Real-time artificial intelligence (AI) systems are structured as a set of tightly coupled layers, including sensing, computing, networking, and application components, which collectively enable end-to-end system functionality. The sensing layer is responsible for acquiring data from physical environments through IoT devices and heterogeneous sensors [1,20]. The computing layer processes this data using edge and cloud-based AI models, balancing latency, accuracy, and resource constraints [4–6]. The networking layer facilitates communication and coordination

across distributed components, leveraging advanced technologies such as 5G and emerging 6G infrastructures [10–11]. At the highest level, the application layer defines system objectives, requirements, and operational constraints, which directly influence the behavior of lower layers [12–13].

These layers are not independent; instead, they exhibit strong interdependencies that significantly affect overall system performance. For instance, increasing sensing resolution enhances data quality but also increases computational load and communication overhead, while network conditions directly influence computation placement and system responsiveness. Application-level requirements further propagate constraints across all layers, requiring coordinated adaptation to meet performance targets.

As a result, optimizing each layer in isolation leads to inefficiencies and suboptimal system behavior. Real-time AI systems require a unified cross-layer approach in which sensing, computation, and communication are jointly optimized under dynamic conditions. This motivates the need for coordinated control mechanisms capable of managing interactions across layers.

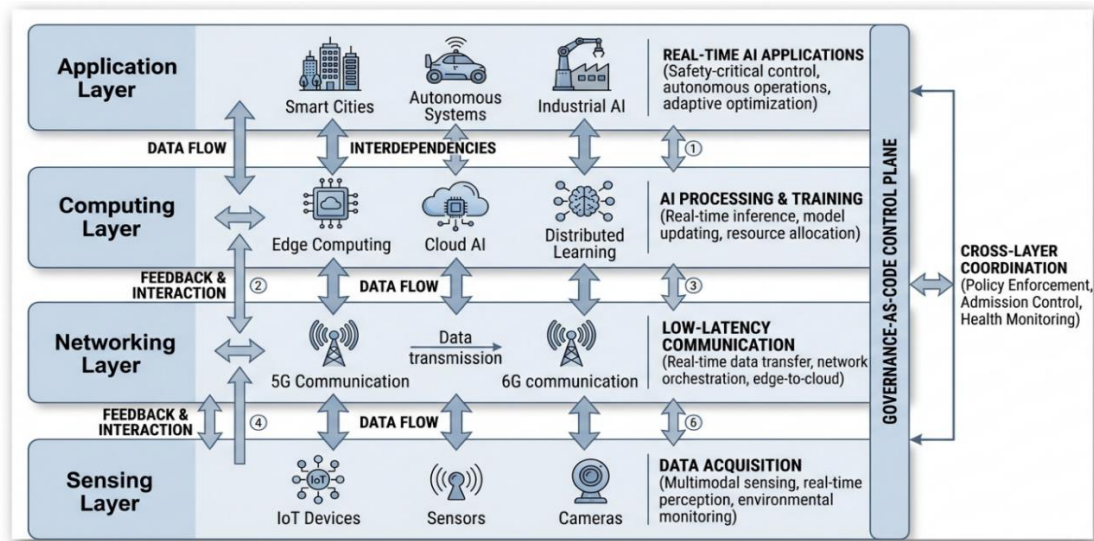


Figure 1. Cross-layer architecture of real-time AI systems showing sensing, computing, networking, and application layers with governance-as-code enabling coordinated control.

The figure illustrates the interaction between sensing, computing, networking, and application layers, highlighting their interdependencies and the flow of data and control signals across the system.

Table 1. Cross-layer components and their functions.

Layer	Description	Key Functions	Challenges
Sensing Layer	Collects data from physical environments using IoT devices and sensors	Data acquisition, signal processing, multimodal sensing	Data noise, energy constraints, trade-off between accuracy and efficiency
Computing Layer	Processes data using edge computing, cloud AI, and distributed learning	Inference, model training, task allocation, edge-cloud collaboration	Limited resources at edge, latency vs accuracy trade-off
Networking Layer	Enables communication	Data transmission, bandwidth	Network variability, congestion, reliability

	between distributed components via 5G/6G technologies	management, latency optimization	issues
Application Layer	Defines system goals and domain-specific requirements	Decision-making, service delivery, system-level optimization	Heterogeneous requirements, real-time constraints
Governance Layer	Cross-layer control mechanism based on policy-driven architecture	Monitoring, policy evaluation, adaptive control, system coordination	Policy design, real-time enforcement, scalability

The table summarizes the roles, characteristics, and interactions of each layer, emphasizing the importance of integrated system design.

3. Governance-as-Code Framework

To address the limitations of fragmented real-time AI systems, we propose a governance-as-code framework based on a two-plane architecture that separates system execution from governance logic. This architecture consists of a data plane and a control plane, which together enable coordinated and adaptive system behavior under dynamic conditions.

The data plane encapsulates the core functional components of the system, including sensing, computing, networking, and application layers. These layers are responsible for data acquisition, processing, communication, and service delivery, forming the operational backbone of real-time AI systems. However, while the data plane provides essential functionality, it does not inherently ensure coordinated behavior across layers.

In contrast, the control plane introduces governance as an intrinsic part of system operation. It incorporates monitoring mechanisms, a policy engine, and adaptation modules that continuously observe system state, evaluate predefined policies, and enforce coordinated actions across layers. This enables real-time decision-making based on system conditions such as latency, workload, and resource utilization.

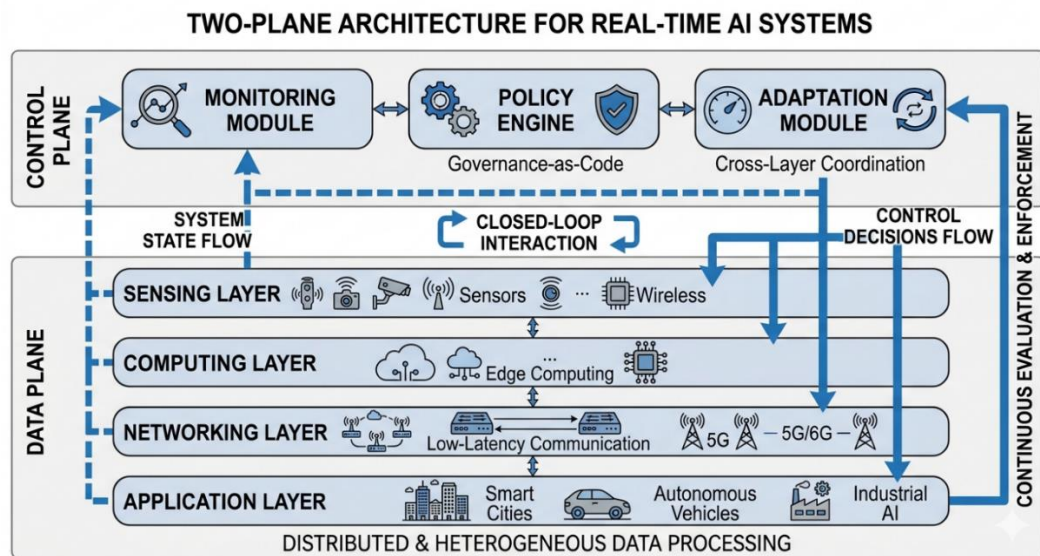


Figure 2. Two-plane governance-as-code architecture showing interaction between data plane (execution) and control plane (policy-driven control) in real-time AI systems.

The interaction between the data plane and control plane forms a closed-loop control process. In this loop, the system continuously monitors its operational state, evaluates policies against current conditions, and applies adaptive actions to maintain optimal performance. Unlike traditional static systems, this approach allows dynamic reconfiguration of system behavior in response to environmental changes.

For example, when network latency increases, the system can dynamically shift computation from cloud to edge nodes while simultaneously reducing sensing resolution to decrease data transmission overhead. Such coordinated actions demonstrate the effectiveness of policy-driven control in maintaining system efficiency and responsiveness.

The figure illustrates the interaction between the data plane and control plane, highlighting the closed-loop process of monitoring, policy evaluation, and adaptive control across system layers.

To further clarify the functional roles of each component, a structured comparison is provided.

Table 2. Data plane vs Control plane comparison.

Aspect	Data Plane	Control Plane (Governance-as-Code)
Primary Role	Executes core system functions	Governs and controls system behavior
Components	Sensing, Computing, Networking, Application	Monitoring, Policy Engine, Adaptation, Control Modules
Functionality	Data acquisition, processing, communication, service delivery	Policy evaluation, decision-making, system coordination
Operation Type	Execution-oriented	Decision and control-oriented
Adaptability	Limited, mostly static or pre-configured	Dynamic and adaptive based on runtime conditions
Awareness	Local (layer-specific)	Global (cross-layer system-wide view)
Control Logic	Embedded within components	Explicit, policy-driven and externally defined
Interaction Style	Sequential or loosely coupled	Coordinated, cross-layer interaction
Response to Change	Reactive and delayed	Proactive and real-time adaptive
Scalability	Limited by resource constraints	Scalable via modular policy updates
Transparency	Low (implicit behavior)	High (explicit and auditable policies)

The table summarizes the responsibilities, key functions, and interactions of the data and control planes, emphasizing the importance of separating execution from governance logic.

4. Research Challenges

Despite significant advances in real-time artificial intelligence (AI) systems, existing approaches remain constrained by several fundamental limitations that hinder their effectiveness in dynamic and distributed environments. One of the primary challenges is the lack of cross-layer coordination, as most current systems optimize sensing, computing, and networking components independently without considering their strong interdependencies. This results in fragmented system behavior, where improvements in one layer may negatively impact overall performance.

Another critical limitation lies in the limited adaptability of current systems under dynamic conditions. Real-time AI applications operate in environments characterized by fluctuating workloads, varying network conditions, and changing resource availability. However, many existing solutions rely on static configurations or predefined assumptions, which restrict their ability to respond effectively to runtime changes.

In addition, there is a clear separation between governance mechanisms and system operation. Traditional governance approaches focus on external processes such as auditing, compliance, and post-hoc evaluation, rather than influencing system behavior during execution. This disconnect prevents real-time enforcement of system-level constraints such as latency, reliability, and safety.

Furthermore, the absence of formal system-level models limits the ability to analyze and optimize real-time AI systems holistically. Current research primarily focuses on component-level improvements, lacking unified abstractions that capture cross-layer interactions and trade-offs. This makes it difficult to design systems that simultaneously satisfy multiple objectives such as performance, efficiency, and reliability.

Finally, existing systems lack policy-driven control mechanisms that enable flexible and dynamic system adaptation. Without explicit policies guiding system behavior, control logic remains embedded within individual components, reducing transparency, flexibility, and scalability.

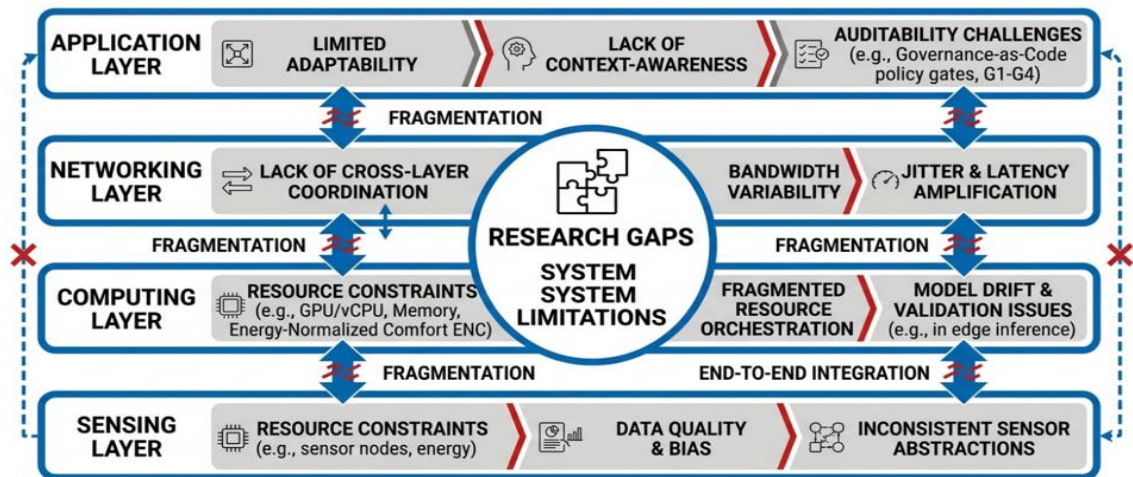


Figure 3. Illustration of key research gaps in real-time AI systems, highlighting the lack of cross-layer coordination, adaptability, and policy-driven control.

The figure highlights key limitations across sensing, computing, networking, and governance dimensions, emphasizing the need for coordinated and policy-driven system design.

Collectively, these challenges demonstrate that current limitations are structural rather than incremental. The proposed governance-as-code approach addresses these issues by embedding governance directly into system runtime, enabling continuous monitoring, adaptive decision-making, and coordinated cross-layer control.

5. Conclusion

This paper introduced governance-as-code as a novel and unifying paradigm for real-time artificial intelligence systems operating in dynamic and distributed environments. By integrating governance mechanisms directly into runtime operation, the proposed approach transforms traditional AI systems from static and fragmented pipelines into adaptive, policy-driven infrastructures.

The presented framework enables continuous monitoring, policy evaluation, and coordinated control across sensing, computing, networking, and application layers. This cross-layer integration allows systems to dynamically respond to changing conditions while maintaining performance, reliability, and efficiency.

Moreover, the proposed approach addresses key limitations of existing systems, including the lack of coordination, limited adaptability, and the disconnection between governance and system operation. By introducing executable policies as a central control mechanism, governance-as-code provides a foundation for flexible, transparent, and scalable system design.

In conclusion, governance-as-code offers a promising direction for the development of next-generation AI systems that are not only intelligent but also adaptive, reliable, and intrinsically governed. This paradigm is particularly relevant for emerging applications in cyber-physical systems, smart infrastructures, and safety-critical environments, where real-time performance and trustworthy operation are essential.

References:

- [1] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013.
- [2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
- [3] J. Ren, Y. Pan, R. Gschossmann, S. Guo, T. Taleb, and M. Chen, "Edge intelligence for the Internet of Things: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2338–2361, Feb. 2021.
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [5] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
- [6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017, pp. 1273–1282.
- [8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020.
- [9] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, 2019.
- [10] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI as a key driver," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

- [11] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, “Artificial neural networks-based machine learning for wireless networks: A tutorial,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.
- [12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Aug. 2014.
- [13] X. Xu, Y. Lu, W. Li, and B. Sheng, “Industry 5.0: A survey on enabling technologies and potential applications,” *Journal of Industrial Information Integration*, vol. 26, 2022.
- [14] D. Amodei et al., “Concrete problems in AI safety,” arXiv:1606.06565, 2016.
- [15] B. H. M. van der Sloot, “The European Union’s Artificial Intelligence Act: A critical analysis,” *Computer Law & Security Review*, vol. 41, 2021.
- [16] NIST, “Artificial Intelligence Risk Management Framework (AI RMF 1.0),” National Institute of Standards and Technology, 2023.
- [17] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [19] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [20] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.