Hyper-Aether: AI-Native Computing Architecture with Virtual Machines as Semantic Computational Units

Hirofumi Inomata
Hyper-Aether Laboratory
inomata@acm.org

Abstract

We present Hyper-Aether, an AI-native computing architecture that unifies dynamic logical systems, graph-based execution, and hardware-level virtualization. We extend this architecture by formalizing virtual machines (VMs) as first-class semantic computational units.

In this framework, computation is modeled as stochastic graph evolution over VM structures, intelligence as a time-evolving formal system, and meaning as value-guided organization of structure and interaction.

Virtual machines are defined as stateful transformers embedded in a dynamically reconfigurable graph. Their granularity, internal computational models, and semantic roles are adaptively determined by an AI-driven control layer.

This work provides a unified foundation for AI-native systems in which execution, reasoning, and meaning are integrated into a single computational paradigm.

1 Introduction

Modern computing systems separate execution and reasoning. Operating systems manage resources and isolation, while AI systems operate at higher abstraction levels.

Hyper-Aether removes this separation. It proposes a unified architecture where:

- virtual machines are the primary execution units,
- system structure is dynamically synthesized,
- control is performed by an AI-driven layer.

Unlike conventional systems, virtual machines are not merely execution environments. They are semantic computational units that contribute directly to meaning formation.

2 Theoretical Foundations

A formal system is defined as:

$$F = (\Gamma, R) \tag{1}$$

Gödel's incompleteness theorem implies:

$$\exists \varphi : \mathrm{true}(\varphi) \land \neg(\Gamma \vdash \varphi) \tag{2}$$

Thus, sufficiently expressive systems cannot be closed. We model evolving systems:

$$A = \{A_t\}_{t \in T}, \quad A_t \subseteq A_{t+1} \tag{3}$$

3 AI as a Dynamic Logical System

We model AI as:

$$M = \{F_t\}_{t \in T} \tag{4}$$

$$F_{t+1} = U(F_t, x_t, \eta_t) \tag{5}$$

where x_t is input and η_t is stochastic variation.

4 Hyper-Aether Architecture

We define:

$$H = (C, V, R, A) \tag{6}$$

- C: compute substrate
- V: VM space
- R: routing network
- A: AI Shell

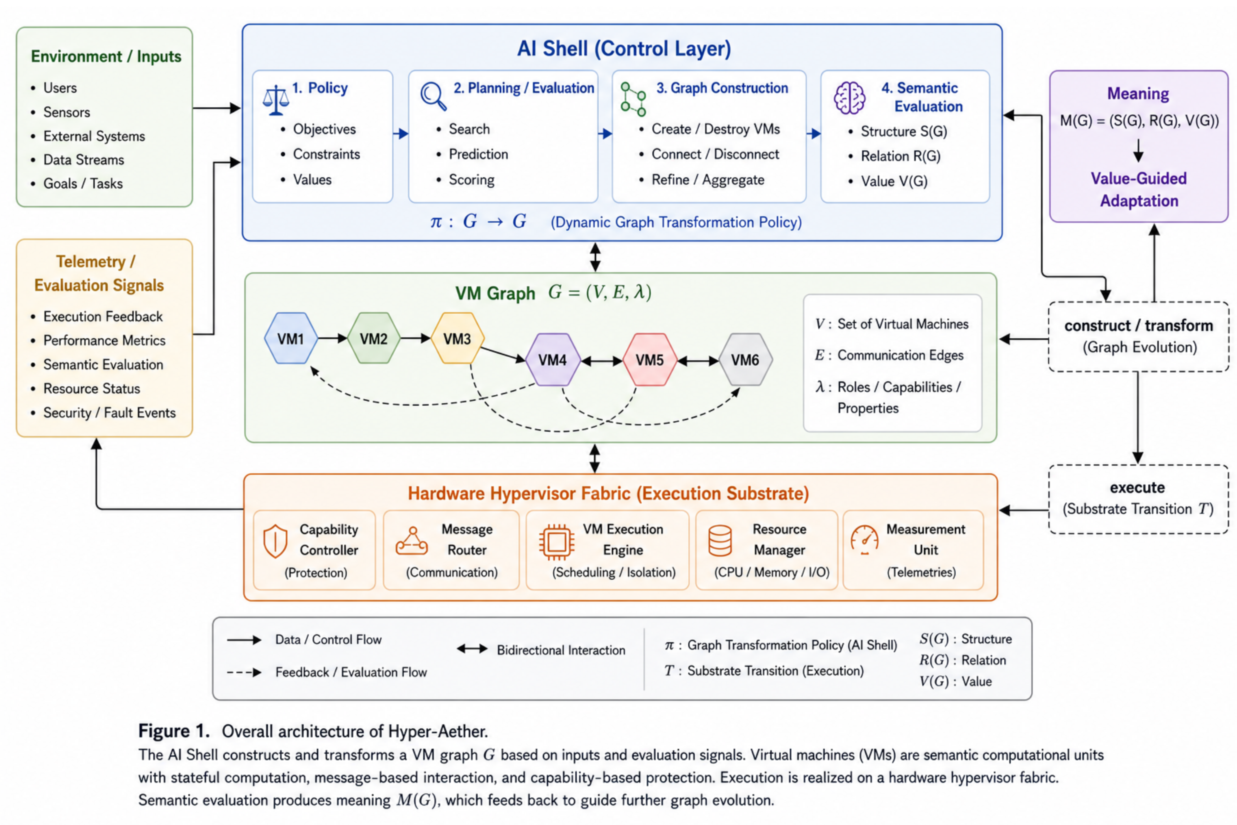

Figure 1: Overall architecture of Hyper-Aether with VM functions.

5 VM as a Computational Primitive

A VM is defined as:

$$s' = f(s, x), \quad y = g(s, x) \tag{7}$$

Thus, a VM is a stateful transformer with message-based interaction.

6 Graph-Based Execution Model

The system is represented as:

$$G = (V, E, \lambda) \tag{8}$$

Execution is:

$$G_{t+1} = T(G_t, \pi_t) \tag{9}$$

7 VM Granularity

VM granularity is adaptive:

- coarse-grained (applications)
- medium-grained (functional roles)
- fine-grained (rules/functions)

Transformations include splitting and merging.

8 Internal Computational Models

VMs support heterogeneous models:

- symbolic systems
- neural networks
- reactive systems

9 Dynamical System Formulation

$$\Omega = \{G\} \tag{10}$$

$$G_{t+1} = \Phi(G_t), \quad \Phi = T \circ \pi \tag{11}$$

With stochasticity:

$$G_{t+1} \sim \Phi(G_t, \eta_t) \tag{12}$$

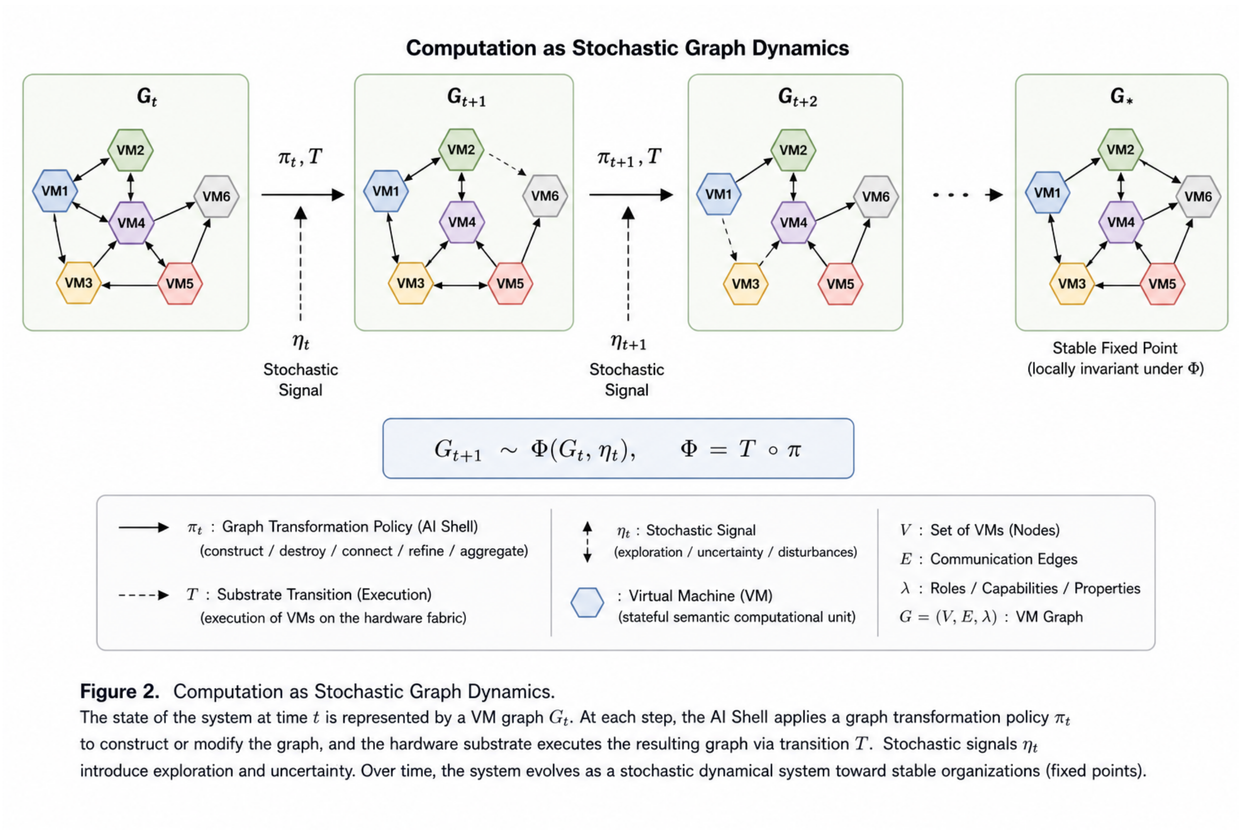

Figure 2. Computation as Stochastic Graph Dynamics.
The state of the system at time t is represented by a VM graph G_t. At each step, the AI Shell applies a graph transformation policy π_t to construct or modify the graph, and the hardware substrate executes the resulting graph via transition T. Stochastic signals η_t introduce exploration and uncertainty. Over time, the system evolves as a stochastic dynamical system toward stable organizations (fixed points).

Figure 2: Computation as stochastic graph dynamics.

10 Semantics and Meaning

$$M(G) = (S(G), R(G), V(G)) \tag{13}$$

VMs contribute to:

- structure
- relation
- value

11 Creativity

$$G^* = \arg\max_{G \in \Omega} V(G) \tag{14}$$

Creativity arises from guided exploration of graph space.

12 Category-Theoretic Interpretation

Let $\mathcal{C}_G$ be a category:

- objects: VMs

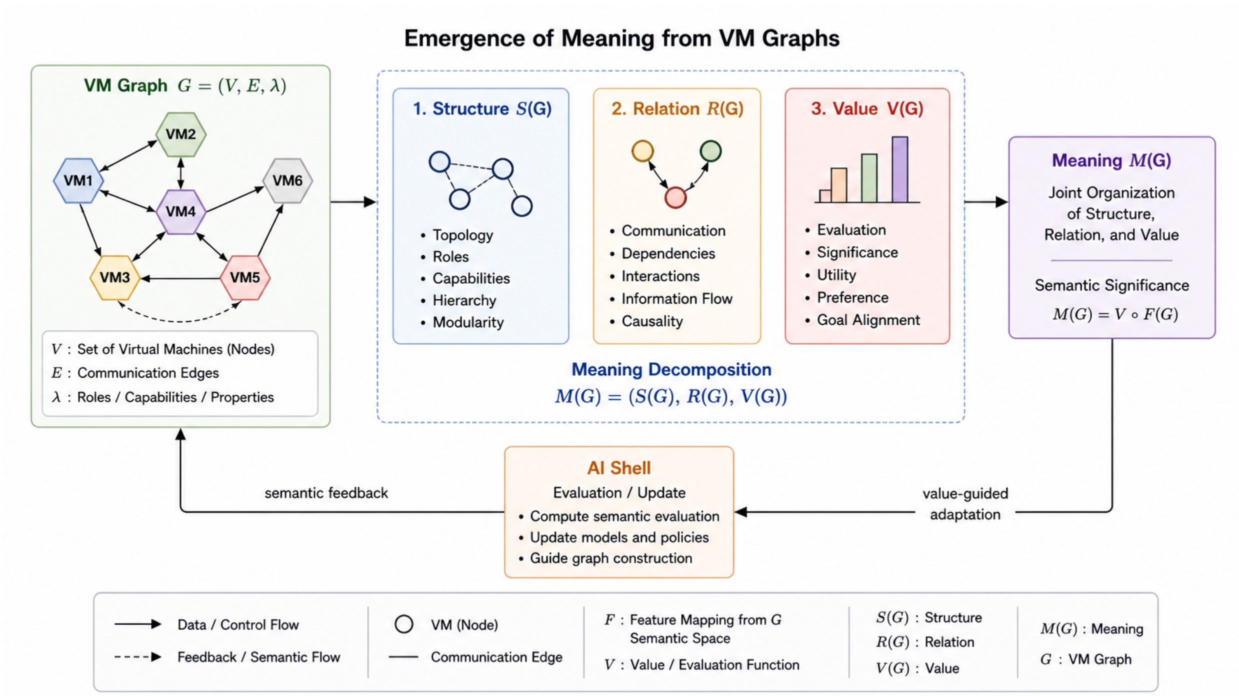

Figure 3. Emergence of Meaning from VM Graphs.
Meaning is not determined by structure alone, but by the joint organization of structure $S(G)$, relation $R(G)$, and value $V(G)$. These components are integrated into semantic significance $M(G)$. The AI Shell evaluates semantic significance and feeds it back to guide the construction and transformation of VM graphs. Through this closed loop, the system adapts toward more meaningful configurations.

Figure 3: Emergence of meaning from VM graphs.

- morphisms: communication

$$\pi : \mathcal{C}_G \to \mathcal{C}_G \tag{15}$$

13 Expressiveness

If VMs are Turing complete:

$$\forall f \in \text{Computable}, \exists G : G \models f \tag{16}$$

14 Advantages

- reduced trusted computing base
- hardware-level isolation
- semantic-native execution
- adaptive graph structure

15 Limitations

- hardware complexity
- verification difficulty
- value function definition

16 Future Work

- VM granularity optimization
- semantic evaluation learning
- hardware prototype

17 Conclusion

Hyper-Aether unifies computation, reasoning, and meaning.

Key claim:

> Computation is the evolution of VM graphs, and virtual machines are the minimal units of meaning, execution, and control.