ED and Computation: How Physical Becoming Implements Information Processing

Allen Proxmire
February 2026

Abstract

Computation has traditionally been defined in terms of its substrates—bits in classical systems, amplitudes in quantum systems. In the ED ontology, these substrates are not fundamental. The universe is made of **event density** and **gradients of becoming**, and computation arises when these gradients are shaped, stabilized, and coupled into reliable patterns of transformation. This paper develops the physical foundation of computation in an ED universe. It shows how ED motifs replace bits, how coherent gradient bundles replace qubits, and how boundary layers, saturation wells, and mobility channels form the architectural primitives of ED‑native computation. Computation is presented not as symbolic manipulation or quantum evolution, but as **pattern‑shaping in the dynamics of becoming**—the controlled evolution of ED motifs through engineered gradients. This physical regime forms the substrate on which symbolic, procedural, and algorithmic architectures (developed in ED‑19) ultimately depend. The paper concludes by outlining the technological implications of ED computation, including temporal engineering, stability engineering, horizon manipulation, self‑modifying architectures, and the fundamental limits imposed by the structure of becoming itself.

1. Introduction — Why Computation Must Be Re‑Founded in ED

Computation has always been defined in terms of its substrates. Classical computation assumes bits and circuits. Quantum computation assumes amplitudes and unitary evolution. Both inherit their architectures from the physical theories beneath them. If the underlying ontology changes, the meaning of computation must change with it.

In the ED ontology, the universe is not made of particles, fields, or wavefunctions. It is made of **event density** and **gradients of becoming**—continuous, dynamical structures whose evolution generates the phenomena we interpret as matter, energy, information, and agency. In such a universe, computation cannot be the manipulation of discrete symbols, nor the evolution of quantum amplitudes. These are architectural layers that emerge much later. At the physical level, computation is the **controlled shaping of ED gradients**.

This paper develops the physical foundation of computation in an ED universe. It shows how computation emerges from the dynamics of event density, how ED gradients implement information flow, and how mobility, saturation, and boundary layers become the primitives of physical computation. It identifies what replaces bits and qubits, what an "ED processor" is, and how computation becomes possible in a substrate where becoming is continuous, dynamical, and architecturally structured.

Symbolic computation, as developed in ED‑19, is an architectural regime built on top of this substrate. It is not the root of computation but one of its higher‑order expressions. The physical layer described here is the foundation on which symbolic, procedural, and algorithmic architectures ultimately depend.

In ED terms:
Computation is the controlled evolution of event‑density motifs through gradients of becoming.

This is the physical meaning of computation—the first layer of procedural structure in a universe whose ontology is dynamical from the ground up.

2. The ED Substrate as a Computational Medium

Computation requires a substrate capable of carrying, transforming, and stabilizing patterns. In classical physics, this substrate is matter arranged into circuits. In quantum physics, it is amplitudes evolving under unitary dynamics. In the ED ontology, the substrate is neither material nor quantum in the conventional sense. It is **event density**—a continuous field of becoming whose gradients determine how structure evolves.

To understand computation in an ED universe, we must understand how event density behaves, how gradients propagate, and how mobility, saturation, and boundary layers create the conditions for information processing. Computation is not imposed on this substrate; it is an architectural consequence of its dynamics.

2.1 Event Density as the Fundamental State Variable

Event density (ED) is the universe's primary state variable. It encodes how much becoming is occurring in a region and how tightly that becoming is coupled to neighboring regions. Local ED configurations are not particles or fields; they are **motifs of becoming**—structured patterns that can persist, interact, and transform.

In ED terms:
A "state" is a local configuration of event density.

These configurations are the physical carriers of information.

2.2 Gradients as Transformation Potentials

Where event density varies, gradients appear. Gradients determine:

- how motifs propagate
- how fast they evolve
- how strongly they interact
- which transformations are possible

A gradient is not a force; it is a **direction of becoming**. Computation emerges when gradients are shaped so that motifs evolve in controlled, predictable ways.

In ED terms:
A gradient is a potential for transformation.

2.3 Mobility and Saturation

Two properties govern how ED motifs behave:

- **Mobility** — how easily a region of ED can reconfigure
- **Saturation** — how resistant a region is to change

High mobility enables rapid transformation.
High saturation stabilizes patterns against disruption.

Computation requires both:

- mobility for operations
- saturation for memory

Their interplay is the physical dialectic that makes information processing possible.

2.4 Boundary Layers

Where gradients steepen or collide, **boundary layers** form. These layers:

- reflect motifs
- refract them
- couple them
- annihilate or merge them
- redirect their trajectories

Boundary layers are the ED substrate's natural control structures. They are the physical precursors of logic gates, routing channels, and coupling interfaces.

In ED terms:
A boundary layer is a region where the dynamics of becoming become architecturally structured.

3. What Replaces Bits and Qubits

Classical computation is built on bits.
Quantum computation is built on qubits.

Both assume that information is encoded in discrete, well-defined states of a physical system.

In the ED ontology, discreteness is not fundamental. It is an emergent property of **saturation**, **stability**, and **boundary formation**. The substrate itself is continuous becoming. Yet this continuity does not prevent computation; it enables a richer form of it. What replaces bits and qubits are **ED motifs**—structured configurations of event density whose stability, mobility, and coupling allow them to function as information-bearing units.

This section identifies the physical primitives of ED-native computation.

3.1 Bits → ED Motif States

A classical bit is a stable, discrete state of a physical system.

In ED, the analogue is a **stable ED motif**—a localized configuration of event density that:

- maintains its identity across time
- resists small perturbations
- can be distinguished from neighboring motifs
- can participate in controlled transformations

These motifs are not particles or symbols. They are **stability wells in the dynamics of becoming**. Their discreteness is emergent: a saturated region behaves as if it has "snapped" into a recognizable state.

In ED terms:
A bit is a saturated ED motif with two stable configurations.

3.2 Qubits → Coherent Gradient Bundles
Quantum computation relies on superposition and coherence.

In ED, the analogue is a **coherent gradient bundle**—a region where multiple transformation potentials overlap in a synchronized way.

A coherent gradient bundle:

- supports multiple potential evolutions simultaneously
- maintains phase‑like relationships through mobility synchronization
- can be steered by shaping boundary layers
- collapses into a stable motif when saturation dominates

Superposition is reinterpreted as **coexisting gradient potentials**.
Coherence is **synchronized mobility** across a region of ED.

In ED terms:
A qubit is a coherent bundle of ED gradients whose evolution can be shaped before saturation resolves it.

3.3 Gates → Gradient‑Coupling Events
Logical and quantum gates are operations that transform states.

In ED, operations arise from **gradient‑coupling events**—interactions where ED motifs:

- merge
- reflect
- refract
- annihilate
- or redirect one another

These interactions are not imposed externally. They are the natural consequences of how gradients meet. By engineering boundary layers and mobility channels, one can shape these interactions into reliable computational primitives.

In ED terms:
A gate is a controlled interaction between ED motifs mediated by engineered gradients.

3.4 Registers → Stability Wells
Memory in classical and quantum systems requires stable storage.

In ED, stability arises from **saturation wells**—regions where event density is high enough to resist change.

A stability well:

- holds an ED motif in place
- preserves its identity across time
- prevents unwanted gradient propagation
- acts as a physical anchor for information

These wells are not containers. They are **regions of slowed becoming**—local minima in the mobility landscape.

In ED terms:
A register is a saturated region that stabilizes an ED motif against transformation.

4. The Architecture of ED‑Native Computation

Computation is not the existence of states alone, nor the presence of gradients, nor the stability of motifs. It is the **organization** of these elements into a system capable of producing controlled, repeatable, and structured transformations. In classical and quantum computation, this organization is provided by circuits, gates, and control logic. In the ED ontology, it is provided by the **dynamics of event density**—how motifs evolve, how gradients interact, how boundary layers shape flow, and how saturation stabilizes patterns.

This section develops the physical architecture of computation in an ED universe. It mirrors the structure of ED‑19, but at the ontological layer rather than the symbolic or procedural layer. The same architectural roles appear—state, operation, control, memory—but their physical meaning is entirely different.

4.1 State as ED Configuration

A computational state is a snapshot of the system's information-bearing structure. In ED, a state is not a discrete symbol or a quantum amplitude vector. It is a **local configuration of event density**—a pattern of becoming that encodes constraints on future evolution.

An ED state is defined by:

- the distribution of event density
- the shape and steepness of local gradients
- the mobility profile of the region
- the presence or absence of saturation wells
- the arrangement of boundary layers

These features determine what transformations are possible, how fast they occur, and how stable the resulting motifs will be.

In ED terms:
A state is a configuration of ED motifs that constrains the system's next step in becoming.

4.2 Operations as Gradient‑Driven Transformations

Operations are the transformations that move the system from one state to another. In ED, operations are not symbolic rules or unitary matrices. They are **gradient‑driven transformations**—the natural evolution of ED motifs under the influence of local potentials.

An ED operation occurs when:

- gradients propagate
- motifs merge or split
- boundary layers redirect flows
- saturation wells stabilize or release patterns
- mobility shifts allow reconfiguration

These transformations are not imposed externally. They are the intrinsic dynamics of the substrate. Computation arises when these dynamics are **shaped** so that transformations become reliable, repeatable, and architecturally structured.

In ED terms:
An operation is a controlled evolution of ED motifs through engineered gradients.

4.3 Control as Boundary‑Layer Architecture
Control determines which operations occur, when they occur, and how they interact. In ED, control is not a program counter or a classical logic circuit. It is the **architecture of boundary layers**—regions where gradients steepen, reflect, couple, or redirect motifs.

Control emerges from:
- gradient routing channels
- reflective or refractive boundary layers
- mobility gates that open or close pathways
- saturation thresholds that trigger transitions
- coupling interfaces that synchronize motifs

These structures determine the procedural flow of ED computation. They are the physical analogues of branching, looping, sequencing, and synchronization.

In ED terms:
Control is the organization of boundary layers that shapes the procedural unfolding of ED dynamics.

4.4 Memory as Saturation‑Based Persistence
Memory requires persistence across time. In ED, persistence is not stored bits or quantum states. It is **saturation‑based stability**—regions where event density is high enough to resist change and hold motifs in place.

Memory arises when:
- saturation wells trap motifs
- mobility is locally suppressed
- gradients cannot easily penetrate
- patterns become attractors
- stability outcompetes transformation

These regions act as physical registers, caches, and long‑term storage. They are not containers but **slow zones of becoming**—areas where change is inhibited enough to preserve structure.

In ED terms:
Memory is the persistence of ED motifs in saturated regions that resist transformation.

4.5 The Architectural Synthesis
Computation in ED emerges when:
- **states** provide ED configurations
- **operations** transform those configurations
- **control** shapes the flow of transformations
- **memory** stabilizes motifs across time

This is the physical architecture of computation in a universe made of becoming. It is not symbolic, not quantum, and not digital. It is **dynamical**—a system of controlled transformations in the ED substrate.

5. How Computation Emerges from ED Dynamics

Computation does not need symbols, circuits, or amplitudes. It requires only a substrate capable of carrying patterns, transforming them, and stabilizing them long enough for those transformations to matter. In the ED ontology, these capacities arise naturally from the dynamics of event density. Computation is not added to the universe; it is an architectural consequence of how becoming unfolds.

This section shows how computation emerges directly from ED dynamics—how gradients propagate information, how motifs interact, how stability and mobility create the conditions for memory and transformation, and how boundary layers become the physical basis of control. Computation is the **controlled evolution of ED motifs**, and this control is achieved by shaping the substrate's inherent dynamics.

5.1 Gradient Propagation as Information Flow

Information is not a substance. It is a constraint on possible evolution. In ED, these constraints are encoded in **gradients**—differences in event density that determine how motifs can move, interact, and transform.

When a gradient propagates:

- it carries structural information
- it influences the evolution of neighboring motifs
- it restricts or expands possible transformations
- it shapes the future trajectory of the region

Information flow is simply **gradient flow**.
Computation begins when gradient flow is **shaped**.

In ED terms:
Information is the pattern of constraints encoded in ED gradients.

5.2 Pattern-Shaping as the Core Operation

In symbolic computation, the core operation is rule application.
In quantum computation, it is unitary evolution.

In ED computation, the core operation is **pattern-shaping**—the controlled evolution of ED motifs through engineered gradients.

Pattern-shaping includes:

- amplifying or suppressing motifs
- redirecting flows through boundary layers
- merging or splitting gradient bundles
- stabilizing motifs in saturation wells
- triggering transitions when thresholds are crossed

These transformations are not symbolic. They are **dynamical**.

Computation is the art of shaping these dynamics so that they produce reliable, repeatable outcomes.

In ED terms:
Computation is the controlled shaping of ED motifs through the dynamics of becoming.

5.3 Stability vs. Mobility as the Computational Dialectic

Every computational system must balance two opposing demands:

- **Mobility** — the capacity to change
- **Stability** — the capacity to persist

In ED, these are not abstract properties. They are physical features of the substrate:

- Mobility arises from low saturation and high gradient responsiveness.
- Stability arises from high saturation and low mobility.

Computation requires:

- mobility for operations
- stability for memory
- controlled transitions between the two

This dialectic is the physical foundation of all information processing.

Where mobility dominates, the system computes.
Where saturation dominates, the system remembers.

In ED terms:
Computation is the interplay between mobility‑driven transformation and saturation‑driven persistence.

5.4 Boundary Layers as Natural Logic

Boundary layers are regions where gradients steepen, collide, or reorganize. They are the ED substrate's natural control structures. When motifs encounter a boundary layer, they may:

- reflect
- refract
- merge
- annihilate
- synchronize
- redirect

These interactions are the physical analogues of:

- logic gates
- routing channels
- synchronization primitives
- conditional branching
- interference and coherence control

Boundary layers are not added to the system. They emerge wherever gradients interact. Computation becomes possible when these layers are **engineered**—when the substrate is shaped so that motif interactions become predictable and architecturally structured.

In ED terms:
Boundary layers are the physical basis of control in ED computation.

5.5 The Emergence of Computation

Computation emerges in ED when:

- gradients propagate information
- motifs evolve through pattern-shaping
- stability and mobility form a dialectic
- boundary layers provide control
- saturation wells provide memory

Nothing symbolic is required.
Nothing quantum is required.
Nothing digital is required.

Computation is the **proceduralization of becoming** at the physical layer.

6. What an "ED Processor" Is

If computation in an ED universe is the controlled evolution of event-density motifs, then an **ED processor** is any region of spacetime where those evolutions can be shaped, stabilized, and coupled in reliable ways. It is not a machine in the classical sense, nor a quantum device in the conventional sense. It is an **architected zone of becoming**—a region where gradients, boundary layers, and saturation wells are arranged so that ED motifs undergo predictable transformations.

An ED processor is not built from components.
It is built from **conditions**—from the arrangement of mobility, saturation, and gradient structure.

It is a *procedural region* rather than a physical object.

This section defines the architecture of such a region.

6.1 Definition

An **ED processor** is a region of the ED substrate where:

- gradients can be shaped
- motifs can be stabilized
- boundary layers can be engineered
- mobility can be modulated
- interactions can be controlled

so that the evolution of ED motifs becomes **procedural** rather than chaotic.

In ED terms:
An ED processor is a shaped region of becoming that implements controlled pattern-shaping.

6.2 Components of an ED Processor

Although ED processors are not built from hardware, they have structural elements—architectural features of the substrate that play the roles of computational primitives.

6.2.1 Gradient Sources

Regions where ED gradients originate or are injected.
These act as the “inputs” to computation.

6.2.2 Boundary-Layer Gates

Steep gradient interfaces that reflect, refract, or couple motifs.
These are the ED equivalents of logic gates and routing structures.

6.2.3 Saturation Wells

Localized regions of high ED saturation that stabilize motifs.
These serve as registers, caches, and memory anchors.

6.2.4 Mobility Channels

Paths of low saturation and high mobility that allow motifs to propagate.
These are the ED equivalents of wires, buses, and waveguides.

6.2.5 Coupling Interfaces

Regions where motifs interact, synchronize, or merge.
These implement multi-motif operations and coherent transformations.

These components are not objects but **architectural features of the ED landscape**.

6.3 Modes of Operation

An ED processor can operate in several modes, depending on how gradients and boundary layers are arranged.

6.3.1 Sequential Mode

Motifs propagate through a series of boundary layers, each performing a transformation.
This is the ED analogue of classical sequential logic.

6.3.2 Parallel Mode

Multiple motifs evolve simultaneously in different mobility channels.
Parallelism is natural in ED because gradients propagate continuously.

6.3.3 Coherent Mode

Gradient bundles evolve in synchronized fashion, maintaining phase-like relationships.
This is the ED analogue of quantum coherence, but without amplitudes.

6.3.4 Distributed Mode

Computation occurs across a spatially extended region, with motifs interacting non-locally through gradient coupling.
This is the ED analogue of distributed computing and field-based computation.

These modes are not mutually exclusive.

An ED processor can shift between them dynamically.

6.4 Why ED Processors Exceed Classical and Quantum Architectures
ED processors are not limited by the assumptions of classical or quantum computation.

6.4.1 No Discrete Clock
Becoming is continuous; computation unfolds at the rate of local mobility.
Timing is intrinsic, not externally imposed.

6.4.2 No Binary Encoding
Information is encoded in gradients, motifs, and saturation patterns—not in discrete states.
This allows richer representational capacity.

6.4.3 No Wavefunction Collapse
Coherence is mobility synchronization, not amplitude superposition.
There is no measurement problem.

6.4.4 Computation as Continuous Becoming
ED processors compute by shaping the dynamics of the substrate itself.
They are not symbolic machines; they are **architected flows of becoming**.

In ED terms:
An ED processor is a region where the universe computes by shaping its own dynamics.

7. Computational Limits in ED Physics
Every computational architecture inherits its limits from the substrate that implements it. Classical computation is bounded by thermodynamic costs and signal propagation speeds. Quantum computation is bounded by decoherence, error rates, and the structure of unitary evolution. ED computation inherits its limits from the **dynamics of becoming**—from how steep gradients can be, how fast motifs can reconfigure, how stable saturation wells can remain, and how boundary layers behave under extreme conditions.

These limits are not engineering constraints.

They are **ontological constraints**—the structural boundaries of what computation can be in a universe made of event density. This section identifies the fundamental limits that shape ED‑native computation and previews the deeper analysis developed in Paper F.

7.1 Maximum Gradient Steepness — The Speed Limit of Becoming
Computation in ED depends on gradients: they carry information, drive transformations, and shape motif interactions. But gradients cannot steepen arbitrarily. Beyond a certain threshold:
- mobility collapses
- boundary layers destabilize
- motifs lose coherence
- saturation spikes and freezes evolution

This threshold defines the **maximum rate of becoming**—the fastest possible transformation the substrate can support. It is the ED analogue of:

- the speed of light (as a limit on propagation)
- the Margolus–Levitin bound (as a limit on quantum evolution)

In ED terms:
There is a maximum steepness beyond which gradients cannot propagate without collapse.

This sets the upper bound on computational speed.

7.2 Minimum Mobility — When Computation Freezes

Just as gradients cannot steepen indefinitely, mobility cannot drop to zero without consequence. When mobility becomes too low:

- motifs cannot reconfigure
- gradients cannot propagate
- boundary layers become rigid
- computation halts

This is the ED analogue of:

- absolute zero (as a limit on thermal motion)
- decoherence (as a limit on quantum evolution)

In ED terms:
There is a minimum mobility below which computation cannot proceed.

This sets the lower bound on computational activity.

7.3 Stability Collapse — When Memory Fails

Memory in ED depends on saturation wells—regions where becoming slows enough to preserve motifs. But saturation cannot increase indefinitely. Beyond a certain point:

- wells collapse into runaway saturation
- motifs lose identity
- gradients cannot escape
- the region becomes inert

This is the ED analogue of:

- gravitational collapse (as a limit on stability)
- memory corruption (as a limit on information persistence)

In ED terms:
There is a maximum saturation beyond which memory becomes unstable.

This sets the upper bound on storage density and persistence.

7.4 Boundary-Layer Breakdown — When Control Dissolves

Control in ED depends on boundary layers—regions where gradients interact in structured ways. But boundary layers are fragile. Under extreme conditions:

- gradients shear them apart
- mobility spikes disrupt their structure
- saturation wells distort their geometry
- coupling interfaces lose coherence

When boundary layers break down, computation becomes chaotic.

This is the ED analogue of:
- logic gate failure
- decoherence in quantum gates
- turbulence in fluid computation

In ED terms:
There is a limit to how finely boundary layers can be engineered before they lose structural integrity.

This sets the upper bound on computational complexity and precision.

7.5 The Ontological Boundary of Computation
Together, these limits define the **computational horizon** of an ED universe:
- maximum gradient steepness → speed limit
- minimum mobility → activity floor
- maximum saturation → memory limit
- boundary-layer breakdown → control limit

These are not engineering constraints.
They are **structural consequences of becoming**.

Paper F will develop these limits in full, showing how they define:
- the ultimate speed of computation
- the maximum density of memory
- the deepest possible temporal wells
- the strongest possible horizons
- the limits of self-modifying agency

In ED terms:
The limits of computation are the limits of becoming itself.

8. Implications for Technology
If computation is the controlled evolution of event-density motifs, then the engineering of computation is the engineering of becoming. The ED ontology does not merely reinterpret existing technologies; it expands the space of what technology *is*. Classical and quantum devices manipulate matter and amplitudes. ED-native devices manipulate **gradients, saturation, mobility, and boundary layers**—the fundamental structures of physical becoming.

This section outlines the technological implications of ED computation and previews the domains developed in Papers B–F. Each domain is not speculative but architectural: a direct consequence of how event density behaves.

8.1 Temporal Engineering (Preview of Paper B)
If computation depends on the rate of becoming, then controlling that rate becomes a technological primitive. ED allows:

- **temporal acceleration** — increasing local mobility
- **temporal deceleration** — deepening saturation wells
- **temporal wells** — regions where becoming slows dramatically
- **temporal shields** — regions insulated from external gradients

These are not metaphors. They are **gradient‑shaping operations**.
Temporal engineering becomes the ED analogue of clock control, synchronization, and time dilation.

In ED terms:
To engineer time is to engineer mobility.

8.2 Memory and Stability Engineering (Preview of Paper C)
Memory is not storage. It is **stabilized becoming**.

ED enables devices that:

- create long‑lived saturation wells
- lock motifs into stable attractors
- preserve patterns against gradient intrusion
- tune stability without freezing computation

This is the physical foundation of ED‑native memory architectures.

In ED terms:
To store information is to shape saturation.

8.3 Horizon Engineering (Preview of Paper D)
Hawking, Unruh, and de Sitter phenomena are not exotic edge cases.

They are **boundary‑layer effects** in ED.

This means horizons can be engineered:

- artificial Hawking‑like emitters
- Unruh‑like acceleration devices
- de Sitter‑like expansion boundaries
- gradient‑induced thermal interfaces

These become tools for computation, energy extraction, and information flow.

In ED terms:
A horizon is a boundary layer with extreme gradient structure.

8.4 Agency and Self‑Modification (Preview of Paper E)
If computation is pattern‑shaping, then an agent is a system that shapes its own patterns. ED makes this literal:

- agents manipulate their own gradients
- agents reshape their own saturation wells
- agents alter their own mobility profiles
- agents become self‑modifying architectures

This is the physical foundation of autoprocedural systems — the next threshold after computation and recursion.

In ED terms:
Agency is self‑directed gradient shaping.

8.5 The Limits of ED Technology (Preview of Paper F)
The limits of computation are the limits of becoming.

ED imposes fundamental constraints:
- maximum gradient steepness
- minimum mobility
- maximum saturation
- boundary‑layer stability thresholds

These define the ultimate boundaries of:
- computational speed
- memory density
- temporal manipulation
- horizon engineering
- self‑modifying intelligence

In ED terms:
The limits of technology are the limits of the substrate's ability to shape itself.

8.6 The Architectural Horizon
The implications of ED computation are not incremental.

They redefine:
- what a device is
- what information is
- what memory is
- what time is
- what agency is
- what intelligence is

ED computation is the first step in a technological arc that extends from gradient shaping to temporal engineering, horizon manipulation, self‑modifying architectures, and the ultimate limits of physical becoming.

9. Conclusion — Computation as Physical Becoming
Computation is not an abstraction layered onto the universe. It is a structural consequence of how the universe becomes. In the ED ontology, the substrate is not matter, fields, or amplitudes. It is **event density**—a continuous

field of becoming whose gradients determine how patterns evolve. When these gradients are shaped, stabilized, and coupled, computation appears. Not as symbol manipulation. Not as quantum evolution. But as **pattern‑shaping in the dynamics of becoming**.

This paper has shown that computation arises when:

- ED motifs form stable configurations
- gradients propagate information
- boundary layers structure interactions
- saturation wells preserve patterns
- mobility enables transformation
- stability enables memory
- controlled gradient shaping produces reliable procedures

These are not engineering choices. They are **ontological consequences** of the ED substrate. Computation is the first domain where becoming becomes **procedural** at the physical layer—where the universe's own dynamics can be shaped into structured, repeatable transformations.

Symbolic and algorithmic computation, as developed in ED‑19, are higher‑order architectures built on top of this substrate. They are not the root of computation but its **cognitive expression**. ED computation is the physical foundation on which symbolic, procedural, and algorithmic regimes depend.

This foundation opens into a broader technological arc:

- **Temporal engineering** — shaping the rate of becoming
- **Memory and stability engineering** — shaping saturation
- **Horizon engineering** — shaping boundary layers
- **Agency and self‑modification** — shaping one's own gradients
- **Ultimate limits** — the boundaries of what becoming can support

These domains are not speculative. They follow directly from the structure of event density. They define the physical possibilities and constraints of a universe whose ontology is dynamical from the ground up.

In ED terms:
Computation is the ED regime where physical becoming becomes procedural becoming.

It is the hinge between the substrate's dynamics and the architectures that minds, agents, and technologies build upon it.

Paper B will develop the next threshold: how the rate of becoming can be shaped, how temporal tension can be engineered, and how time itself becomes a manipulable dimension of ED physics.