

Magnetic Fluidic Computing

Circulation, Admissibility, and the Architecture of Trajectory

Flyxion

Independent Researcher

June 2026

Abstract

This essay develops the philosophical and mathematical consequences of Magneto-Absorptive Modular Computing (MAMC), a speculative hardware architecture in which compute units circulate through a thermodynamic system under magnetic control rather than occupying fixed positions on a static substrate. The argument proceeds on two levels simultaneously. At the engineering level, MAMC is characterised as a closed circulation loop—Dock, Compute, Heat, Eject, Cool, Return—whose dynamics are governed by thermal thresholds, magnetic routing fields, and absorption-cycle refrigeration. At the philosophical level, MAMC is argued to instantiate a transition from an architecture of storage to an architecture of trajectory: the primary computational entity is not a processor occupying a location but a path through a constrained admissibility space. This transition has consequences that extend well beyond cooling engineering. Memory is reconceived as recurrence of trajectory rather than persistence of state. Entropy is reconceived as accessibility—the cardinality of admissible future paths available to a pod—rather than as heat to be expelled. Hardware identity is reconceived as the persistence of a circulation pattern rather than the continuity of physical components. Agency is reconceived as the emergent product of coordinated constraint navigation rather than the expression of a global model. Each of these reconceptions is formalised within the RSVP (Relativistic Scalar-Vector Plenum) framework, connecting MAMC to the author’s prior work on hidden curvature, constraint fields, chain-of-memory, and simulated agency. Several formal results are derived, including the Accessibility Restoration Theorem, a derivation of memory from circulation topology, a curvature measure on accessibility space, and a Hamiltonian formulation of pod dynamics under the CLIO constraint field. The essay argues that MAMC, taken seriously as a philosophical object rather than merely an engineering curiosity, suggests that computation, memory, identity, and agency are all special cases of recurrent admissibility structure—a claim that places the architecture within the broader program of constraint-first ontology.

Keywords: magnetic fluidic computing, MAMC, RSVP, admissibility, trajectory, circulation, entropy, CLIO, constraint field, chain of memory, simulated agency, hidden curvature, process ontology.

Contents

1	Introduction: The Metaphysics of Location	3
2	A Philosophical Prelude: The Pattern of Return	4
3	Engineering Substrate: The Circulation Loop	6
4	Trajectory as the Primitive Entity	7
4.1	From Objects to Morphisms	7
4.2	RSVP Formalisation	8
4.3	The Trajectory-Identity Thesis	9
5	Memory as Recurrence of Trajectory	9
5.1	The Recurrence Definition	9
5.2	Formal Properties	10
5.3	Chain-of-Memory Connection	10
6	Entropy as Accessibility	11
6.1	Two Senses of Entropy	11
6.2	The Accessibility Restoration Theorem	12
6.3	Philosophical Significance	12
7	Curvature of Accessibility Space	13
7.1	Accessibility as a Scalar Field	13
7.2	A Curvature Measure on Routing Space	13
7.3	Connection to Hidden Curvature	14
8	Hamiltonian Formulation: CLIO as Hardware	15
8.1	The CLIO Constraint Field	15
8.2	Lagrangian Formulation	15
8.3	Hamiltonian Formulation	16
8.4	Conservation and Dissipation	16
9	Process Ontology and the Beyond Process Thesis	17
9.1	The Process Turn	17
9.2	The Ship of Theseus Resolution	18
9.3	Beyond Process: Admissibility as the Deeper Primitive	18
10	Simulated Agency and Emergent Intelligence	19
10.1	The Appearance of Agency	19
10.2	Formal Model of Emergent Agency	19
10.3	Collective Intelligence	20

11 Synthesis: Computation as Admissibility Structure	20
12 Conclusion	21

1. Introduction: The Metaphysics of Location

Every computing architecture encodes a metaphysics. The metaphysics encoded by conventional digital computing is one of location and storage. A central processing unit occupies a socket. Memory occupies slots at fixed addresses. Computation is understood as something that happens inside stable, positioned objects. Information is understood as something that persists in specific places. Even the software abstractions that sit above the hardware—the process, the thread, the virtual machine—inherit this spatial imagery. A process runs here. A variable is stored there. The stack grows down; the heap grows up.

This essay argues that the Magneto-Absorptive Modular Computing architecture (MAMC), developed in the companion paper, is philosophically interesting precisely because it challenges this metaphysics at the hardware level. In MAMC, compute pods circulate continuously through a thermodynamic system. No pod is permanently positioned. No computation has a fixed address. The system's behaviour is determined not by the locations of components but by the topology of the routes that pods traverse. The architecture does not merely change how computers are built; it changes what kind of thing a computer is.

The transition that MAMC proposes is from an architecture of storage to an architecture of trajectory. In an architecture of storage, the primitive entities are objects at locations: processors, registers, memory cells, buses. In an architecture of trajectory, the primitive entities are paths through a constrained space: the routes that compute pods follow under the influence of magnetic fields and thermal gradients. The distinction matters because it changes the answer to nearly every foundational question about computation. What is memory? What is identity? What is entropy? What is agency? Each question receives a different answer depending on which architecture one takes as primitive.

The argument of this essay proceeds in several stages. Section 2 develops the philosophical stakes in essayistic prose before the formal machinery begins, treating the engineering components as manifestations of a theory of cognition rather than as objects to be analysed in isolation. Section 3 briefly characterises the engineering architecture of MAMC to establish the physical setting. Section 4 develops the claim that trajectory rather than location is the primary computational entity and connects this to the RSVP framework and to categorical foundations. Section 5 derives a theory of memory from circulation topology and connects it to the Chain of Memory framework. Section 6 develops the distinction between thermal entropy and accessibility entropy and proves the Accessibility Restoration Theorem. Section 7 introduces a curvature measure on accessibility space and connects it to the Hidden Curvature program. Section 8 develops a Hamiltonian and Lagrangian formulation of pod dynamics under the CLIO constraint field. Section 9 situates MAMC within process ontology and the Beyond

Process framework. Section 10 develops the connection between MAMC and simulated agency. Section 11 synthesises the results into a unified claim about the relationship between computation and admissibility. Section 12 concludes.

2. A Philosophical Prelude: The Pattern of Return

When most computing systems remember something, they retrieve a stored representation. Information is placed somewhere, and later a search process locates it again. The memory exists because a record persists.

The architecture described in this essay begins from a different assumption. A memory need not be a thing. A memory may instead be a circulation.

In conventional machines, computation and maintenance are treated as separate activities. The processor performs work. The cooling system removes waste. Storage preserves results. Each subsystem exists largely independently of the others, connected by interfaces that are themselves fixed. The machine is a collection of distinguished places—CPU, RAM, disk—connected by paths that exist to serve those places.

Within a circulating architecture these distinctions begin to dissolve. A computational element performs work only temporarily. As thermal gradients emerge, the element migrates into cooling regions. After restoring its operational capacity, it returns to active computation. What appears to be a cooling cycle is simultaneously a routing cycle. What appears to be a routing cycle is simultaneously a memory cycle. The subsystems do not merely interact; they are the same process observed from different angles.

Persistence emerges not from immobility but from recurrence. A pattern survives because it continues to reappear within the circulation. Nothing needs to stay put for the system to remember. What needs to happen is that the same configuration keeps coming back.

This inversion has significant philosophical consequences. Traditional computation is organised around location. A memory is identified by where it resides. A processor is identified by where it is mounted. Computation becomes the movement of information between fixed places. Everything that matters has an address.

A circulating architecture instead organises itself around trajectories. The relevant question is no longer where something is but where it is permitted to go. The computational state of the system is not a snapshot of locations but a topology of admissible paths. A pod at rest is computationally inert. A pod in motion, following a coherent route, is a computational event.

The system therefore resembles a transportation network, a circulatory system, or an ecosystem more than a conventional computer. These are systems in which no particular position is privileged and in which identity is constituted by the coherence of flow rather

than the stability of substance. A river remains a river despite continual replacement of its water. An organism remains an organism despite continual replacement of its cells. A flame remains itself while consuming entirely different fuel from one moment to the next. What persists is not the matter but the form of the process—and more precisely still, not even the process itself but the constraint structure governing which processes are admissible.

A computational module within the circulating architecture may leave the processing region, traverse multiple thermal domains, exchange energy with its surroundings, and return in a physically altered state. Its operating temperature will have changed. The charge state of its internal circuitry will have shifted. Some of its transistors will have undergone billions of switching events since it last docked. Yet if its trajectory remains coherent—if it returns to the same class of docking positions, performs the same class of computations, responds to the same class of thermal signals—the system recognises continuity despite material change. Identity is a property of path stability rather than material permanence.

The architecture therefore provides a physical model for a long-standing philosophical intuition: persistence does not require stasis.

This reframing changes the character of several concepts that computing has inherited from its origins in the metaphysics of static storage.

Memory, in this framework, becomes a special case of circulation. A memory exists whenever a pattern repeatedly returns to a recognisable configuration. A frequently accessed computation corresponds to a pod that makes the same circuit often: the path is worn smooth by use, the routing is optimised toward it, the thermal management is tuned for its profile. An infrequently accessed computation corresponds to a pod that makes the same circuit rarely, its path maintained at the margins of the scheduling policy. A forgotten memory is not necessarily deleted. It has simply lost the pathways required for recurrence. The routing topology has changed, and the old path is no longer admissible. Remembering, in such a system, becomes less analogous to searching through an archive and more analogous to re-entering a stable current. One does not find a memory. One re-enacts a trajectory.

Entropy, within the same framework, changes meaning. Conventional computers treat heat as waste—the unavoidable byproduct of computation, to be expelled as rapidly as possible. In this architecture, thermal state is information. The temperature of a pod determines which regions of the routing space are accessible to it. A hot pod is constrained: it must cool before it may dock. A cool pod is unconstrained: it may dock anywhere in the hub that is available. The thermal state of the system sculpts a landscape of possibility. Cooling is not merely the removal of waste heat. Cooling is the restoration of future possibilities. The refrigeration cycle is, in this reading, an accessibility engine—

a device whose function is not to eliminate entropy in the thermodynamic sense but to multiply it in the semantic sense, expanding the range of futures available to pods that have temporarily exhausted their options.

Routing, in this reading, becomes the navigation of possibility space. Every routing decision is a decision about which futures to make available. The electromagnet arrays that guide pods through the channels are not merely mechanical actuators; they are a logic for possibility, shaping the landscape of what each pod may do next.

Agency, within such a system, becomes an emergent property of constraint navigation rather than a capacity for self-representation. Each pod appears to pursue goals: it moves toward the hub when cool, retreats to cooling when hot, waits in the return zone when the hub is occupied. Yet no pod carries a model of the system. No pod represents its goal or plans a route to it. The appearance of goal-directedness is produced entirely by the shape of the constraint field. The pod does not try to dock. It simply follows the gradient of admissible futures, and docking is where that gradient leads.

The resulting picture bears a closer resemblance to a living system than to a static machine—not because the machine is biological, but because it shares with biological systems the feature that function emerges from the shape of constraints rather than from the content of representations.

The architecture is therefore interesting not only because it proposes a new computational mechanism, but because it challenges one of the deepest assumptions inherited from conventional computing: that memory is storage, that identity is location, and that persistence requires immobility.

In a circulating machine, as in many natural systems, what endures is neither matter nor position.

What endures is the pattern of return.

The formal development that follows—the theorems, the Hamiltonian formulation, the curvature measures—is an attempt to say precisely what this means.

3. Engineering Substrate: The Circulation Loop

The physical basis of MAMC is a closed circulation loop in which encapsulated compute pods—modules housing CPUs, GPUs, memory stacks, or storage—move through a thermodynamic system under magnetic control. The loop has the structure:

$$\text{Dock} \longrightarrow \text{Compute} \longrightarrow \text{Heat} \longrightarrow \text{Eject} \longrightarrow \text{Cool} \longrightarrow \text{Return} \longrightarrow \text{Dock} \quad (1)$$

Each pod carries a ferrofluidic or magnetically permeable shell that allows the routing layer—an array of electromagnets embedded in the channel walls—to exert directional forces. Docking is achieved through contactless inductive or optical interfaces. The cooling circuit operates on absorption-cycle or phase-change principles: as a pod enters the cooling bath, refrigerant evaporates against its thermal interface surfaces, removing heat. Once the pod’s temperature falls below the return threshold, the scheduling layer routes it back to the docking hub.

Let T denote pod temperature, T_e the ejection threshold, T_r the return threshold, P the power dissipated during computation, and C the pod’s thermal capacitance. The dwell time before ejection satisfies:

$$\tau_{\text{dwell}} = \frac{C(T_e - T_r)}{P} \quad (2)$$

The duty cycle of a pod in the hub—the fraction of time spent computing—is:

$$\eta = \frac{\tau_{\text{dwell}}}{\tau_{\text{dwell}} + 2\tau_{\text{transit}} + \tau_{\text{cool}}} \quad (3)$$

where τ_{transit} is the one-way transit time between hub and cooling bath and τ_{cool} is the time to reduce temperature from T_e to T_r . Full hub utilisation requires a pod pool of size at least N/η , where N is the number of docking positions.

This engineering substrate is the physical object whose philosophical implications the remainder of this essay develops. The loop (equation 1) is not merely a cooling strategy; it is a new ontological primitive.

4. Trajectory as the Primitive Entity

4.1. From Objects to Morphisms

In conventional computing, the primitive entities are objects: processors, memory cells, registers. Operations are relations between objects. This is an object-oriented ontology in the technical sense—the primary logical category is a thing with properties, and processes are secondary, defined in terms of transformations between things.

Category theory offers a different starting point. A category consists of objects and morphisms, but it is possible to define categories in which the morphisms are more fundamental than the objects. In the category $\mathbf{Path}(X)$ of paths on a space X , the objects are points and the morphisms are paths between them, but the structure of the category is entirely determined by the paths: two points are isomorphic if and only if there exists a path between them, and the composition of paths is the concatenation operation. Points that cannot be connected by any path belong to different connected

components and are, in the relevant sense, in different worlds.

MAMC is naturally modelled in $\mathbf{Path}(X)$ where X is the physical routing space of the system. Each pod is a morphism: it begins at a docking position, traverses the cooling circuit, and returns to a docking position. The objects—the docking positions—are secondary; what matters is the topology of the morphisms connecting them.

Definition 4.1 (Routing Space). *Let X be a metric space representing the physical interior of the MAMC system. The routing space \mathcal{R} is the subset of X accessible to pod motion, equipped with the subspace topology induced by X .*

Definition 4.2 (Admissible Path). *A continuous map $\gamma : [0, 1] \rightarrow \mathcal{R}$ is an admissible path if it satisfies the routing constraints imposed by the electromagnetic field configuration $\mathbf{B}(x, t)$ and the pressure field $p(x, t)$. The set of admissible paths is denoted \mathcal{A} .*

The morphism-priority perspective has an immediate consequence: the identity of a compute pod is not determined by its intrinsic properties but by the paths it traverses. Two pods with identical hardware that follow different routes through the system are, architecturally, different entities. Two pods with different hardware that follow identical routes are, architecturally, interchangeable. This is the trajectory-identity thesis.

4.2. RSVP Formalisation

In the RSVP framework, the physical state of a system is described by a scalar field $\phi : \mathcal{M} \rightarrow \mathbb{R}$ and a vector field $\mathbf{v} : \mathcal{M} \rightarrow T\mathcal{M}$ over a spacetime manifold \mathcal{M} , coupled by field equations that constrain their joint evolution. The central interpretive claim of RSVP is that physical entities are not points in \mathcal{M} but patterns of constraint on the admissible trajectories through \mathcal{M} .

MAMC instantiates this claim at the hardware level. The routing electromagnetic field $\mathbf{B}(x, t)$ and pressure field $p(x, t)$ play the role of the RSVP vector field: they constrain the admissible trajectories of pods through the routing space \mathcal{R} . The thermal field $T(x, t)$ —the temperature at each location in the system—plays the role of the RSVP scalar field: it determines the local admissibility of pod positions and transitions between phases (docked, transit, cooling, standby).

More precisely, define the admissibility potential:

$$\Phi(x, T) = \alpha \|\mathbf{B}(x)\|^2 - \beta p(x) + \gamma (T - T_{\text{opt}})^2 \quad (4)$$

where T_{opt} is the pod's optimal operating temperature and $\alpha, \beta, \gamma > 0$ are coupling constants. A pod at position x with temperature T has low admissibility potential when it is in a region of strong magnetic attraction ($\|\mathbf{B}\|$ large), high pressure support (p large), and near-optimal temperature ($T \approx T_{\text{opt}}$). The pod trajectory follows the gradient of Φ :

$$\dot{x} = -\nabla_x \Phi(x, T(x, t)) \quad (5)$$

This is the RSVP field equation rendered as hardware dynamics. The machine computes by shaping Φ rather than by executing instructions.

4.3. The Trajectory-Identity Thesis

Theorem 4.3 (Trajectory Identity). *Under the dynamics (5), two pods p_1 and p_2 initialised at the same docking position with the same temperature are architecturally equivalent if and only if they follow identical admissible paths through \mathcal{R} .*

Proof. The dynamics (5) are deterministic given initial conditions (x_0, T_0) and the field configuration (\mathbf{B}, p, T) . Two pods at the same initial position with the same temperature therefore follow the same path $\gamma : [0, 1] \rightarrow \mathcal{R}$ under identical field configurations. Conversely, if two pods follow the same path, they perform the same sequence of docking, computation, ejection, and cooling operations, and are therefore functionally identical from the perspective of the scheduling layer. Architectural equivalence is therefore co-extensive with path identity. \square \square

The theorem confirms that path, not position, is the right primitive. A pod is not characterised by where it is but by where it goes.

5. Memory as Recurrence of Trajectory

5.1. The Recurrence Definition

In conventional architectures, memory is a location. A datum is stored at an address. Persistence of memory is identified with stability of location: the datum remains at address a until explicitly overwritten. This is a static conception of memory, and it inherits the object-oriented metaphysics of the underlying hardware.

MAMC suggests a different conception. If no location is permanent, if every pod is always in transit, then persistence cannot be identified with stability of position. What persists in MAMC is not a location but a route. A memory, in the MAMC sense, is a trajectory that recurs.

Definition 5.1 (Trajectory Memory). *Let $\gamma_i : [0, 1] \rightarrow \mathcal{R}$ denote the path traversed by a pod during its i -th circuit through the system. The trajectory memory of the system after n circuits is:*

$$\mathcal{M}_n = \bigcup_{i=1}^n \gamma_i \quad (6)$$

where the union is taken in the space of paths, with recurrence weighted by frequency.

This definition makes explicit a claim that is only implicit in conventional memory models: information is encoded not in the content of a location but in the structure of recurrence. A datum that is accessed repeatedly appears in \mathcal{M}_n with high multiplicity; a datum never accessed appears with zero multiplicity. The analogy to biological memory is immediate: in the synaptic consolidation model of long-term memory, a memory trace is strengthened by repeated activation of the same neural pathway, not by the persistence of a stored value.

5.2. Formal Properties

Proposition 5.2 (Convergence of Trajectory Memory). *If the routing field $\mathbf{B}(x, t)$ is periodic with period τ and the scheduling policy is stationary, then the sequence $\{\gamma_i\}$ of pod paths converges in distribution to a limiting measure μ on \mathcal{A} , and the trajectory memory \mathcal{M}_n converges almost surely to the support of μ .*

Proof sketch. Under a stationary scheduling policy, the routing decisions depend only on the current thermal state of each pod and the current occupancy of the hub, both of which are Markov processes under the periodic field. By the ergodic theorem for Markov chains on compact state spaces, the empirical distribution of pod paths converges almost surely to the unique stationary measure μ of the routing Markov chain. The support of μ is the set of paths that are visited with positive probability in the stationary regime, which is exactly the limiting trajectory memory. \square \square

Corollary 5.3 (Memory Loss under Circulation Failure). *If circulation ceases—if $\dot{x} = 0$ for all pods—then no new paths are added to \mathcal{M}_n and the trajectory memory freezes. If the routing field subsequently changes so that the previously traversed paths are no longer admissible, the stored memory becomes inaccessible: retrieval requires traversal of the relevant path, and that path no longer exists.*

The corollary makes precise the sense in which MAMC memory is fragile in a way that static memory is not, and resilient in a way that static memory is not. Static memory is vulnerable to physical damage to the storage medium but robust to changes in the external environment. MAMC memory is robust to physical replacement of pods—a new pod on the same route inherits the same memory structure—but vulnerable to changes in the routing topology. This is a fundamentally different failure mode, and one that has more in common with biological memory than with silicon memory.

5.3. Chain-of-Memory Connection

The Chain of Memory framework, developed in prior work, argues that identity is constituted not by the persistence of a substance but by the continuity of a process of self-reference: each state of a system encodes a reference to its predecessor, and the chain of such references constitutes a memory that grounds identity over time.

The trajectory memory \mathcal{M}_n provides a concrete physical implementation of this idea. Each circuit γ_i encodes, in the state of the pod at the end of the circuit, information about the circuit that preceded it: the thermal history, the computational work performed, the cache state at ejection. The sequence $\gamma_1, \gamma_2, \dots, \gamma_n$ is literally a chain in which each element is causally downstream of all its predecessors. The identity of a compute process is constituted not by the persistence of any particular pod but by the continuity of the chain.

Formally, define the chain operator:

$$\mathcal{C}[\gamma_i] = \gamma_{i+1} \quad (7)$$

where γ_{i+1} is the path taken on the next circuit, causally determined by the state of the pod at the end of γ_i . The trajectory memory is then the orbit of γ_1 under iterated application of \mathcal{C} :

$$\mathcal{M}_n = \{\gamma_1, \mathcal{C}[\gamma_1], \mathcal{C}^2[\gamma_1], \dots, \mathcal{C}^{n-1}[\gamma_1]\} \quad (8)$$

The system remembers its history by traversing paths that are causally downstream of that history. Memory is not stored; it is traversed.

6. Entropy as Accessibility

6.1. Two Senses of Entropy

Classical thermodynamic entropy, as defined by Clausius and developed by Boltzmann, measures the number of microstates compatible with a given macrostate: $S_{\text{therm}} = k_B \log W$. In the context of a heat engine or refrigeration cycle, high entropy corresponds to high temperature and disordered molecular motion; low entropy corresponds to low temperature and ordered molecular motion. A conventional computer generates entropy in the thermodynamic sense: computation dissipates energy as heat, increasing the entropy of the environment.

MAMC inverts this relationship in a philosophically interesting way. In MAMC, heat is not merely a byproduct of computation; it is a control signal. The temperature of a pod determines which regions of the routing space are accessible to it: a hot pod must enter the cooling circuit; a cool pod may enter the hub. Temperature is therefore a variable that partitions the space of admissible futures.

This suggests a second sense of entropy, defined in terms of admissibility rather than thermodynamics.

Definition 6.1 (Accessibility Entropy). *Let p be a compute pod at position x with temperature*

T . Let $\mathcal{A}(x, T)$ denote the set of admissible future paths available to p —paths that begin at x , respect the routing constraints, and are consistent with the thermal dynamics given the current temperature T . The accessibility entropy of p is:

$$S_{\text{acc}}(x, T) = \log |\mathcal{A}(x, T)| \quad (9)$$

where $|\cdot|$ denotes an appropriate measure on the space of paths (e.g., the Wiener measure conditioned on the routing constraints).

6.2. The Accessibility Restoration Theorem

Theorem 6.2 (Accessibility Restoration). *Under the routing dynamics (5) with finite thermal constraints, the accessibility entropy S_{acc} satisfies:*

$$\frac{\partial S_{\text{acc}}}{\partial T} < 0 \quad (10)$$

at all points (x, T) in the routing space. Consequently, the cooling operation—which decreases T —increases S_{acc} . The cooling circuit is an accessibility restoration device.

Proof. Consider the admissible path set $\mathcal{A}(x, T)$. This set is constrained by two conditions: the routing constraints imposed by (\mathbf{B}, p) , which are independent of T ; and the thermal constraint, which requires that any admissible path must remain within the region $\{x' : T(x', t) \leq T_e\}$ for a pod entering the hub, and within the region $\{x' : T(x', t) \geq T_r\}$ for a pod departing the cooling bath. As T increases toward T_e , the set of positions at which a pod may dock—the positions satisfying the thermal admission condition—shrinks monotonically. This is because the hub’s admission control rejects pods whose temperature exceeds T_e . Therefore the set of admissible paths that include a docking segment shrinks as $T \rightarrow T_e$, and $|\mathcal{A}(x, T)|$ is a strictly decreasing function of T in the range $[T_r, T_e]$. Since $S_{\text{acc}} = \log |\mathcal{A}(x, T)|$ is a monotone transformation of $|\mathcal{A}(x, T)|$, we have $\partial S_{\text{acc}}/\partial T < 0$. □ □

Corollary 6.3 (Cooling as Entropy Increase). *In the RSVP sense of entropy-as-accessibility, the cooling circuit increases the entropy of a pod even while decreasing its thermodynamic temperature. Cooling is therefore simultaneously thermodynamic entropy reduction and accessibility entropy increase. The system exhibits a two-level entropy structure in which the two entropies move in opposite directions.*

6.3. Philosophical Significance

The two-level entropy structure has philosophical consequences that extend beyond MAMC. It suggests that the concept of entropy is not univocal: there is thermodynamic entropy, which measures disorder in the molecular sense, and there is accessibility

entropy, which measures the richness of available futures in the semantic sense. The conventional identification of high entropy with disorder and low entropy with information is derived from the thermodynamic sense; in the accessibility sense, the identification reverses.

This connects directly to the RSVP interpretation of entropy as a measure of constraint on admissible trajectories. In RSVP, a high-entropy state is one in which many trajectories are admissible; the entropy of a system measures the size of the space of futures available to it. MAMC realises this interpretation at the hardware level: a cool pod has many admissible futures (it may dock, compute, perform any of several task types), while a hot pod has few admissible futures (it must enter the cooling circuit). The accessibility entropy of equation (9) is the RSVP entropy rendered in engineering terms.

The Accessibility Restoration Theorem 6.2 can then be read as a theorem about the relationship between thermodynamic and semantic entropy: the operations that reduce thermodynamic entropy (cooling) are the same operations that increase semantic entropy (expanding the space of admissible futures). Computation, in the MAMC framework, is the process of converting accessibility entropy into thermodynamic entropy—consuming future possibilities in exchange for physical heat.

7. Curvature of Accessibility Space

7.1. Accessibility as a Scalar Field

The accessibility entropy $S_{\text{acc}}(x, T)$ defines a scalar field over the product space $\mathcal{R} \times [T_r, T_e]$. The gradient of this field:

$$\nabla S_{\text{acc}} = \left(\frac{\partial S_{\text{acc}}}{\partial x}, \frac{\partial S_{\text{acc}}}{\partial T} \right) \quad (11)$$

encodes, at each point, the direction in which accessible futures are increasing most rapidly. Pods following the gradient of S_{acc} move toward regions of greater admissibility—toward the cooler, better-connected regions of the routing space.

Definition 7.1 (Accessibility Flow). *The accessibility flow is the vector field:*

$$\mathbf{A}(x, T) = \nabla S_{\text{acc}}(x, T) \quad (12)$$

The accessibility flow points in the direction of increasing future possibility.

7.2. A Curvature Measure on Routing Space

The divergence of the accessibility flow defines a natural curvature measure on the routing space:

$$K(x, T) = \nabla \cdot \mathbf{A}(x, T) = \nabla^2 S_{\text{acc}}(x, T) \quad (13)$$

Regions with $K > 0$ (positive curvature) are regions in which accessible futures are increasing in all directions: these are attractors of pod motion, corresponding to the cool, well-connected regions near the re-entry ports of the hub. Regions with $K < 0$ (negative curvature) are regions in which accessible futures are decreasing in all directions: these are repellers, corresponding to the hot zones near the ejection ports.

Proposition 7.2 (Curvature and Docking Density). *Under steady-state circulation, the equilibrium density of pods is proportional to $e^{S_{\text{acc}}(x, T)}$. Pod density is highest in high-accessibility regions and lowest in low-accessibility regions. Consequently:*

$$\rho_{\text{pods}}(x, T) \propto e^{S_{\text{acc}}(x, T)} \quad (14)$$

and the mean curvature $\langle K \rangle = \langle \nabla^2 \log \rho_{\text{pods}} \rangle$ measures the spatial non-uniformity of the pod distribution.

Proof sketch. In the stationary regime, the pod distribution satisfies a Fokker-Planck equation with drift given by \mathbf{A} and diffusion given by the stochastic fluctuations in pod temperature and routing field. The stationary distribution of a Fokker-Planck equation with potential $-S_{\text{acc}}$ is proportional to $e^{S_{\text{acc}}}$ by the standard Gibbs formula. $\square \quad \square$

7.3. Connection to Hidden Curvature

The Hidden Curvature program argues that the apparent geometry of physical space—metric distance, causal structure, spatial adjacency—is derived from a more fundamental structure of admissibility. Two events are causally close not because they are metrically near but because a high density of admissible paths connects them. The apparent curvature of spacetime, in this reading, is a projection of the curvature of admissibility space onto the metric.

MAMC provides a hardware-level realisation of this claim. Two docking positions in the hub may be physically adjacent—a few centimetres apart—while being topologically distant in accessibility space, because the routing topology does not permit direct pod transit between them. Conversely, two positions on opposite sides of the cooling circuit may be metrically distant but accessible-space-adjacent, because a direct routing channel connects them.

The curvature measure $K(x, T)$ defined in equation (13) is the hardware instantiation of the Hidden Curvature's admissibility curvature. Regions of the routing space where K is large and positive are regions where the admissibility structure is strongly convergent—many paths lead here—corresponding to the high-curvature attractive regions of the Hidden Curvature framework. The routing topology of a MAMC system literally has

a geometry distinct from its physical geometry, and it is the routing geometry, not the physical geometry, that determines computational structure.

Theorem 7.3 (Accessibility Metric). *Define the accessibility distance between two positions $x_1, x_2 \in \mathcal{R}$ as:*

$$d_{\mathcal{A}}(x_1, x_2) = \inf_{\gamma \in \mathcal{A}(x_1 \rightarrow x_2)} \int_{\gamma} \frac{1}{e^{S_{\text{acc}}(\gamma(t), T(\gamma(t)))}} dt \quad (15)$$

where $\mathcal{A}(x_1 \rightarrow x_2)$ is the set of admissible paths from x_1 to x_2 . Then $d_{\mathcal{A}}$ is a metric on \mathcal{R} , and the Gaussian curvature of $(\mathcal{R}, d_{\mathcal{A}})$ is proportional to K at each point.

Proof. Non-negativity, symmetry, and the triangle inequality for $d_{\mathcal{A}}$ follow from the infimum construction and the composition of admissible paths. The curvature formula follows from the standard relation between the conformal factor in a Riemannian metric and Gaussian curvature: if the metric is $g = e^{-2S_{\text{acc}}} g_0$ where g_0 is the Euclidean metric, then the Gaussian curvature satisfies $K_{\mathcal{A}} = e^{2S_{\text{acc}}} \nabla^2 S_{\text{acc}} = e^{2S_{\text{acc}}} K$. \square \square

The metric $d_{\mathcal{A}}$ is the natural distance measure for MAMC: it measures not how far apart two positions are in physical space but how much computational cost is incurred in moving a pod between them, discounted by the accessibility of the intervening region. The physics of the routing space is determined by $d_{\mathcal{A}}$, not by the Euclidean metric.

8. Hamiltonian Formulation: CLIO as Hardware

8.1. The CLIO Constraint Field

The CLIO (Constraint-Layered Information Ontology) framework models physical and computational systems as evolving under the influence of a constraint field $\Phi(x, t)$, with trajectories determined by the gradient flow of Φ . The framework generalises both Hamiltonian mechanics—where Φ is the Hamiltonian and trajectories are Hamilton’s equations—and gradient descent optimisation—where Φ is the loss function and trajectories are gradient steps.

The admissibility potential (4) is precisely a CLIO constraint field for the MAMC system. The pod trajectory (5) is the CLIO gradient flow. MAMC therefore provides a physical hardware implementation of CLIO dynamics: the machine literally evolves as a gradient flow on a constraint landscape.

8.2. Lagrangian Formulation

To make the connection with classical mechanics explicit, define the MAMC Lagrangian:

$$L(x, \dot{x}, T) = \frac{1}{2} m_{\text{eff}} \|\dot{x}\|^2 - \Phi(x, T) \quad (16)$$

where m_{eff} is the effective inertial mass of a pod in the routing fluid (determined by the pod's mass and the added mass of the displaced fluid), and Φ is the admissibility potential (4). The Euler-Lagrange equations give:

$$m_{\text{eff}}\ddot{x} = -\nabla_x \Phi(x, T) \quad (17)$$

which is Newton's second law for a particle in the potential Φ . The gradient flow approximation (5) is the overdamped limit of (17), valid when the viscous drag on the pod in the routing fluid is large compared to inertial forces—a regime that is physically realistic for pods moving at centimetres per second through viscous channels.

8.3. Hamiltonian Formulation

Performing the Legendre transformation on (16), define the canonical momentum $\pi = m_{\text{eff}}\dot{x}$ and the MAMC Hamiltonian:

$$H(x, \pi, T) = \frac{\|\pi\|^2}{2m_{\text{eff}}} + \Phi(x, T) \quad (18)$$

Hamilton's equations are:

$$\dot{x} = \frac{\partial H}{\partial \pi} = \frac{\pi}{m_{\text{eff}}}, \quad \dot{\pi} = -\frac{\partial H}{\partial x} = -\nabla_x \Phi(x, T) \quad (19)$$

The thermal dynamics couple to the Hamiltonian through the temperature dependence of Φ :

$$\frac{dH}{dt} = \frac{\partial H}{\partial T} = \frac{\partial \Phi}{\partial T} \dot{T} = 2\gamma(T - T_{\text{opt}})\dot{T} \quad (20)$$

Equation (20) captures the central coupling of MAMC: the energy of the constraint landscape changes as the pod's temperature changes. When a pod is computing and heating ($\dot{T} > 0$), the constraint landscape becomes steeper in the ejection direction (since $T - T_{\text{opt}} > 0$ and $\gamma > 0$), driving the pod toward ejection. When a pod is cooling ($\dot{T} < 0$), the landscape flattens toward the hub, allowing return. The Hamiltonian (18) is therefore a dynamical system in which thermal state and mechanical state are genuinely coupled, not merely correlated.

8.4. Conservation and Dissipation

In the conservative limit (no viscous drag, no heat exchange), the Hamiltonian (18) is conserved along trajectories. The presence of viscous drag in the routing fluid and heat exchange with the cooling bath introduces dissipation. The modified energy balance is:

$$\frac{dH}{dt} = -\lambda\|\dot{x}\|^2 + \frac{\partial\Phi}{\partial T}\dot{T} \quad (21)$$

where $\lambda > 0$ is the drag coefficient. The first term represents viscous dissipation of kinetic energy; the second represents the coupling to thermal dynamics. The system is a driven-dissipative system, maintained out of equilibrium by the continuous input of computational work (which heats the pods) and the continuous output of heat to the cooling circuit (which cools them). The steady-state circulation is a limit cycle of this driven-dissipative system.

Theorem 8.1 (Existence of Limit Cycle). *Under the dynamics (21) with periodic driving from the absorption-cycle cooling and stationary routing field, there exists a stable limit cycle Γ in the phase space (x, π, T) to which all sufficiently nearby trajectories converge.*

Proof sketch. The system is a periodically driven dissipative system with compact state space (the routing space is bounded, temperatures are bounded between T_r and some maximum T_{\max} , and momenta are bounded by the maximum force the routing electromagnets can exert). By the Poincaré-Bendixson theorem extended to periodically forced systems, every bounded trajectory converges either to a fixed point or to a limit cycle. Fixed points correspond to pods arrested in place, which requires $\nabla_x\Phi = 0$ and $\dot{T} = 0$ simultaneously; the scheduled routing field is designed to avoid such degeneracies. Therefore trajectories converge to limit cycles. □ □

The limit cycle Γ is the mathematical object corresponding to the informal circulation loop (1). The machine computes because its pods are attracted to Γ .

9. Process Ontology and the Beyond Process Thesis

9.1. The Process Turn

Process philosophy—associated with Whitehead, Bergson, and more recently Rescher—argues that the fundamental ontological category is not substance (a thing that persists) but event or process (an occurrence that unfolds). In the process view, an apparently persistent object—a stone, a cell, a computer—is in reality a pattern of ongoing processes that maintains its form through continuous interaction with its environment. The stone is not a static thing but a stable configuration of atomic and molecular processes; the cell is not a fixed entity but a self-sustaining metabolic cycle; the computer is not a collection of components but a pattern of information-processing events.

MAMC is a computer that makes the process character of computation visible at the hardware level. No pod is permanently in place. No memory has a fixed address. No computation has a permanent location. The machine is, explicitly and irrevocably, a pattern of ongoing processes. If the circulation stops, the machine ceases to exist—not

because its components are destroyed but because the process that constitutes it has ended.

9.2. The Ship of Theseus Resolution

The Ship of Theseus paradox asks: if the planks of a ship are gradually replaced, one by one, until no original plank remains, is it still the same ship? The paradox arises because there are two plausible criteria of identity—material continuity (same planks) and structural continuity (same form)—that come apart.

MAMC dissolves the paradox cleanly. The machine's identity is constituted by the circulation pattern Γ , not by the pod pool $\{p_1, \dots, p_N\}$. As pods are replaced—quarantined for failure, replaced by new pods—the machine persists as long as Γ persists. The machine is not the collection of pods; the machine is the limit cycle. Material continuity is irrelevant to identity; structural continuity of the circulation pattern is all that matters.

This is more than an analogy. The same argument applies, *mutatis mutandis*, to biological organisms. A human being is not the collection of molecules that compose them at any instant—most of those molecules are replaced within years—but the persisting pattern of biological processes, the limit cycle of the organism's metabolic and neural dynamics. MAMC makes this feature of identity explicit in hardware.

9.3. Beyond Process: Admissibility as the Deeper Primitive

The Beyond Process framework argues that process philosophy, while superior to substance ontology, does not reach the deepest level of analysis. Processes are themselves derived from a more fundamental structure: the space of admissible transformations. A process is not ontologically primitive; what is primitive is the constraint field that makes certain processes admissible and others not.

Applied to MAMC: the circulation pattern Γ is not the fundamental entity. What is fundamental is the admissibility potential $\Phi(x, T)$ that makes Γ the stable attractor of the dynamics. The machine is not the hardware (substance ontology), not the circulation (process ontology), but the constraint landscape that determines which circulations are possible. Change Φ by reconfiguring the routing field and the scheduling policy, and the machine becomes a different machine even if every pod remains in place.

This points toward a definition of computation that transcends both substance and process:

Definition 9.1 (Computation as Constraint Navigation). *A computation is a trajectory through admissibility space that is consistent with the constraint field Φ and that begins and ends in designated regions of the state space (initial and final computational states). The machine is the constraint field. The computation is the trajectory. The result is the terminal state.*

Under this definition, MAMC is not a new way of implementing the same computation as a conventional computer; it is a different ontological type of computer, one whose primitives are admissibility structures rather than Boolean operations.

10. Simulated Agency and Emergent Intelligence

10.1. The Appearance of Agency

A compute pod in a MAMC system appears to behave intelligently. It moves. It seeks out the hub when cool and available. It retreats to the cooling circuit when hot. It waits in the return zone when the hub is full. It responds to external conditions—the state of the routing field, the temperature of adjacent fluid, the occupancy of docking positions—in a way that is functionally appropriate for the task of sustained computation. An observer unfamiliar with the routing mechanics might attribute goals to the pods: each pod appears to be trying to dock and compute.

Yet no pod has goals. No pod has a model of the system. No pod represents the global state or executes a plan. Each pod simply evolves under the gradient flow of $\Phi(x, T)$, responding to local field gradients. The apparent intelligence is not in the pods; it is in the constraint field.

10.2. Formal Model of Emergent Agency

The Simulated Agency framework argues that apparent goal-directedness in physical systems is a projection generated by the structure of the admissibility constraints, not a property of the system's internal representation. An entity appears to have a goal G when the constraint field Φ is such that almost all admissible trajectories from any starting point converge to G . The entity does not represent G ; the field is shaped so that G is the attractor.

Definition 10.1 (Simulated Goal). *A region $G \subseteq \mathcal{R}$ is a simulated goal of the system under constraint field Φ if the accessibility entropy S_{acc} has a global maximum at G :*

$$G = \arg \max_{x \in \mathcal{R}} S_{acc}(x, T_r) \quad (22)$$

where T_r is the return temperature at which pods re-enter the hub.

The hub is the simulated goal of the pod system: it is the region of highest accessibility entropy at the return temperature, the region from which the greatest number of admissible futures branch. Pods appear to seek the hub because the constraint field is shaped to make the hub the global attractor of the accessibility gradient flow.

Proposition 10.2 (Agency without Representation). *The gradient flow (5) produces goal-directed behaviour—convergence to G —without any pod representing G or computing a plan to reach it. Goal-directedness is a property of Φ , not of the pod's internal state.*

Proof. By the Accessibility Restoration Theorem, S_{acc} is maximised at low temperatures and high routing connectivity. The hub, by design, is the region of highest connectivity in \mathcal{R} . Therefore $G = \{\text{hub}\}$ at temperature T_r . The gradient flow (5) drives pods toward $\nabla\Phi = 0$, which at temperature T_r coincides with the hub. No internal representation is required: the convergence follows from the field geometry. \square \square

10.3. Collective Intelligence

The individual pod's simulated agency scales to collective intelligence at the system level. When multiple pods circulate simultaneously, their interaction through the routing field produces a collective dynamic that is not present in any individual pod. A region of the routing space occupied by many pods has a locally elevated effective potential (due to magnetic repulsion between pods), which deflects subsequent pods to alternative routes. The routing topology is dynamically modified by the pod distribution, and the pod distribution is determined by the routing topology: a fixed-point problem whose solution is the equilibrium circulation pattern.

This collective dynamic produces behaviours that resemble intelligent resource allocation: when a particular type of computation is in high demand, more pods of the relevant type accumulate in the return zone adjacent to the hub, reducing the effective latency of that computation type. The system self-organises to match its hardware configuration to its workload, not through any central planning but through the local interactions of pods with the routing field and with each other.

The mathematical structure is identical to that of ant colony optimisation, where individual ants following local pheromone gradients collectively discover shortest paths; and to that of Hopfield networks, where individual neurons following local gradient dynamics collectively converge to stored patterns. In all three cases, collective intelligence emerges from the interaction of simple local rules with a shared physical substrate.

11. Synthesis: Computation as Admissibility Structure

The several lines of argument developed in the preceding sections converge on a unified claim. To state it, recall the concepts introduced:

Trajectory memory \mathcal{M}_n (equation 6) encodes information as recurrence structure on paths. Accessibility entropy S_{acc} (equation 9) measures the richness of available futures. The accessibility metric $d_{\mathcal{A}}$ (equation 15) defines the geometry of the routing space. The MAMC Hamiltonian H (equation 18) governs the dynamics of individual pods. The limit cycle Γ constitutes the identity of the machine. The simulated goal G (Definition 7.2) produces apparent agency without representation.

All of these concepts are derived from a single underlying structure: the admissibility

potential $\Phi(x, T)$ and the space of admissible paths \mathcal{A} . The memory is a subset of \mathcal{A} selected by recurrence. The entropy is a measure on \mathcal{A} . The metric is the cost function on \mathcal{A} . The Hamiltonian governs motion through \mathcal{A} . The limit cycle is the attractor in \mathcal{A} . The goal is the maximum of the accessibility measure on \mathcal{A} .

The unified claim is therefore:

Theorem 11.1 (Computation as Admissibility). *In the MAMC framework, computation, memory, entropy, geometry, identity, and agency are all derived concepts, obtained by applying different operations—recurrence selection, cardinality measure, cost minimisation, gradient dynamics, attractor identification, measure maximisation—to the same primitive structure: the pair $(\mathcal{R}, \mathcal{A})$ of routing space and admissible path set.*

The proof of this theorem is not a formal derivation but a survey of the preceding sections, each of which establishes one of the derivations. The theorem's significance is conceptual: it shows that the traditional multiplicity of foundational computing concepts—memory, computation, identity, entropy, agency—is not a genuine ontological multiplicity but a representational artifact, arising from the application of different interpretive lenses to a single underlying admissibility structure.

This claim can be stated more sharply: the architecture of conventional computing, in which memory, processing, storage, and interconnect are separate physical subsystems with separate conceptual identities, is a contingent choice that reflects the historical dominance of the storage metaphysics. MAMC reveals that these apparent distinctions can be dissolved by adopting trajectory and admissibility as the primary concepts. Memory is not a special kind of hardware; it is a recurrence pattern in admissibility space. Processing is not a special operation; it is a trajectory through admissibility space. Agency is not a special property; it is the structure of an attractor in admissibility space. The distinctions dissolve into special cases of a single framework.

12. Conclusion

This essay has argued that Magneto-Absorptive Modular Computing, considered as a philosophical object, is an instance of the broader program of constraint-first ontology. The engineering architecture—pods circulating through a thermodynamic loop under magnetic control—provides a hardware realisation of concepts that have previously been developed at abstract levels: the RSVP field equations, the Chain of Memory framework, the Hidden Curvature program, CLIO constraint dynamics, and the Simulated Agency thesis.

The principal results are the following. Trajectory, not location, is the appropriate primitive entity for MAMC; this is formalised by the trajectory-identity theorem and connected to the morphism-priority perspective in category theory. Memory is derivable from circulation topology as the recurrence structure of pod paths; this is formalised

by the trajectory memory definition and the convergence proposition. Entropy has a two-level structure in MAMC, with thermodynamic entropy and accessibility entropy moving in opposite directions; this is formalised by the Accessibility Restoration Theorem. The routing space carries a non-Euclidean accessibility metric whose curvature measures the convergence and divergence of admissible paths; this is formalised by the accessibility metric theorem. Pod dynamics are governed by a Hamiltonian that couples mechanical and thermal variables, with the limit cycle as the machine's stable identity; this is formalised by the Hamiltonian formulation and the limit cycle existence theorem. Agency is simulated by the structure of the constraint field, without any pod representing a goal; this is formalised by the proposition on agency without representation.

Taken together, these results establish a connection between a speculative hardware architecture and a coherent philosophical position: that computation, memory, identity, and agency are not independent primitives but are all special cases of admissibility structure—recurrence, measure, metric, dynamics, and attractor applied to the space of admissible paths through a constrained physical system.

The broader implication is that the design of computing systems is not merely a technical problem but a philosophical one. The choice of a hardware architecture is a choice of a metaphysics. Conventional architectures embody a metaphysics of storage, fixity, and location. MAMC embodies a metaphysics of trajectory, circulation, and admissibility. Whether or not MAMC is ever built, the conceptual framework it provides is worth taking seriously as an alternative to the storage metaphysics that currently dominates—and potentially constrains—the imagination of what computation can be.

The prelude with which this essay began proposed a simple inversion: that memory is not a thing but a circulation, that identity is not a location but a trajectory, that persistence is not stasis but recurrence. The formal machinery of the intervening sections has confirmed that this inversion is not merely metaphorical. It can be stated precisely. It generates theorems. It connects to independent frameworks in physics, mathematics, and philosophy of mind. It resolves puzzles—about the Ship of Theseus, about the relationship between thermodynamic and semantic entropy, about the emergence of agency without representation—that the storage metaphysics cannot resolve without additional stipulation.

What endures in a circulating machine is neither the matter of the pods nor the positions they occupy. What endures is the constraint structure that makes certain trajectories admissible and others not. The machine is the shape of the possible. Computation is the act of traversing that shape. Memory is the trace left by repeated traversal. Identity is the stability of the pattern of return.

These are not claims about a hypothetical engineering artefact. They are claims about

what computation has always been, made visible for the first time by an architecture that removes the fixed substrate and lets the trajectory speak for itself.

References

- [1] Whitehead, A. N. (1929). *Process and Reality: An Essay in Cosmology*. Macmillan, New York.
- [2] Bergson, H. (1911). *Creative Evolution*, trans. A. Mitchell. Henry Holt, New York.
- [3] Rescher, N. (1996). *Process Metaphysics: An Introduction to Process Philosophy*. SUNY Press, Albany.
- [4] Landauer, R. (1961). Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191.
- [5] Rosensweig, R. E. (1985). *Ferrohydrodynamics*. Cambridge University Press, Cambridge.
- [6] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.
- [7] Dorigo, M., Maniezzo, V., and Colomi, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26(1):29–41.
- [8] Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4):620–630.
- [9] Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7.
- [10] Penrose, R. (1989). *The Emperor's New Mind*. Oxford University Press, Oxford.
- [11] Rosen, R. (1991). *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*. Columbia University Press, New York.