

# Persistence Before Efficiency

A Constraint-First Theory of Civilization

Flyxion

January 22, 2026

## Abstract

Contemporary systems are widely optimized for efficiency, speed, and throughput, yet increasingly exhibit fragility, depletion, and loss of human and ecological capacity. This essay argues that these failures arise from a recurrent architectural error: the prioritization of fast, legible optimization variables over slow, regenerative substrates such as attention, learning, ecological health, and material sufficiency. Through examples drawn from logistics, food distribution, authorization systems, digital platforms, and historical practice, the essay shows how apparent efficiency often reflects cost displacement rather than true resource minimization. When symbolic abstractions such as prices, metrics, and permissions are treated as more real than the physical and cognitive processes they represent, systems undermine their own foundations. The essay proposes a constraint-first design framework in which persistence precedes optimization, emphasizing upper bounds on accumulation, lower bounds on access to essentials, stable interfaces, redundancy, and local regeneration loops. The central claim is that civilizations fail not by lacking efficiency, but by allowing efficiency to destroy the conditions that make continued operation possible.

# 1 Introduction

Modern civilization is widely described as efficient, advanced, and technologically optimized. Yet alongside unprecedented throughput and connectivity, there is a growing sense of fragility: ecological depletion, cognitive overload, supply-chain brittleness, loss of skill, and widespread dependence on systems whose failure would be catastrophic. These outcomes are often treated as accidental side effects of progress, or as political failures external to technical design.

This essay argues instead that these failures arise from a single, recurring structural error: the systematic prioritization of short-term efficiency over long-term persistence.

Efficiency, understood as the maximization of throughput, speed, or output under fixed conditions, is a local optimization criterion. Persistence, by contrast, concerns whether a system can continue to function across time while preserving the substrates it depends upon. When efficiency is allowed to override persistence, systems may appear successful in the short run while undermining their own foundations.

This pattern recurs across domains, including global logistics systems optimized for speed rather than resilience, food systems optimized for export value rather than nourishment, software platforms optimized for engagement rather than learning, and authorization systems optimized for transactional legibility rather than material sufficiency.

The resulting failures are not primarily moral or ideological. They are architectural. They reflect a failure to impose constraints that protect slow, regenerative processes from fast, extractive ones.

The central claim of this essay is simple:

*Any civilization that allows fast optimization processes to consume slow regenerative processes will eventually destroy the conditions that made optimization possible.*

The remainder of the essay develops this claim by distinguishing fast and slow variables, examining how modern systems misclassify them, and outlining design principles that prioritize persistence over efficiency.

## 2 Fast and Slow Variables

To understand the failure mode under discussion, it is necessary to distinguish between fast and slow variables. Fast variables are those that change rapidly, are easy to measure, remain legible to centralized systems, and respond quickly to incentives. Typical examples include transaction volume, engagement metrics, shipping speed, quarterly profit, and authorization signals.

Slow variables, by contrast, change gradually, require long time horizons to evaluate, are highly context-dependent, and resist straightforward quantification. They regenerate only under relatively stable conditions. Examples include soil fertility, ecosystem health, human attention span, skill accumulation, institutional memory, and social trust.

Crucially, slow variables form the substrate upon which fast variables operate. Fast variables cannot exist independently; they draw down, reorganize, or consume slow variables in order to function.

A system is well-designed when fast variables are constrained so that they do not irreversibly degrade the slow variables they depend on. A system is poorly designed when fast variables are allowed to optimize freely, treating slow variables as inexhaustible or irrelevant.

This misclassification is the root of the failures examined in the sections that follow.

### **3 The Illusion of Efficiency**

Modern systems routinely describe themselves as efficient. Goods are delivered rapidly across continents, information flows instantly, and production is highly specialized. These achievements are typically framed as evidence of progress. However, efficiency in this context often reflects not true resource minimization, but cost displacement.

Efficiency becomes illusory when a system appears to reduce cost only by externalizing it in space, time, or social responsibility.

#### **3.1 Efficiency as Cost Displacement**

Consider the production of a simple object such as a wooden chair. In many contemporary supply chains, wood may be harvested in one region, shipped across the globe for processing, assembled elsewhere, and then transported again to the point of use. From a purely financial perspective, this may appear efficient if labor and regulatory costs are lower in intermediate locations.

From a physical perspective, however, the process is energetically absurd. The transformation performed is minimal: atoms are rearranged into a new configuration. The dominant cost is not manufacturing, but transport.

Such arrangements are viable only because fossil fuel energy is subsidized and its costs externalized, environmental damage is excluded from accounting frameworks, labor protections vary widely across regions, and long-term ecological and social costs are systematically ignored.

The system does not eliminate costs; it hides them. What appears as efficiency is, in fact, the exploitation of unpriced slow variables.

## 3.2 Transport as a Misclassified Resource

In historical systems, distance imposed natural constraints. Transport was slow, expensive, and visible. As a result, production tended to remain close to use, and supply chains were shaped by geography.

Modern logistics systems have treated transport as a permanently cheap and abundant resource. Distance has been effectively erased from decision-making. This reclassification allows systems to behave as though spatial separation has no consequence.

The result is not efficiency, but fragility. When transport is disrupted by fuel shortages, geopolitical conflict, natural disasters, or infrastructure failure the system has no local redundancy. Production and consumption have been decoupled to such a degree that even short interruptions produce cascading failures.

## 3.3 Specialization Without Persistence

Specialization is often cited as a driver of efficiency. While specialization can increase productivity under stable conditions, it becomes dangerous when it erodes general capability and local competence.

Highly specialized systems tend to concentrate knowledge within narrow roles, reduce cross-domain understanding, increase dependence on complex coordination mechanisms, and eliminate fallback skills that would otherwise support adaptation under stress.

When specialization is optimized for speed and throughput rather than persistence, the system loses its ability to adapt. Skills that are not exercised decay. Tools that change rapidly cannot be mastered. Local repair knowledge disappears.

The result is a system that performs well only within a narrow operating envelope and fails catastrophically outside it.

## 3.4 Efficiency Versus Survivability

True efficiency must be evaluated across time. A process that maximizes output in the present at the expense of future capability is not efficient in any meaningful sense; it is extractive. Survivability, by contrast, depends on the presence of redundancy, buffering capacity, operational slack, stable interfaces, and explicit limits on extraction rates. These features often appear inefficient when assessed using short-term performance metrics. However, they are precisely the properties that allow systems to persist, adapt, and recover under conditions of stress.

Modern optimization regimes routinely remove these features in the name of efficiency. The resulting systems are brittle, overfit to present conditions, and unable to recover from disturbance.

### 3.5 Summary

The illusion of efficiency arises when systems optimize fast, legible variables while ignoring the slow substrates that make those variables meaningful. Transport, specialization, and scale become tools of cost displacement rather than genuine improvement.

In the next section, this pattern will be examined in a domain where the inversion is especially stark: authorization systems governing access to basic necessities.

## 4 Authorization Over Reality

Modern societies increasingly govern access to essential goods not through physical availability or human need, but through symbolic authorization systems. Food, tools, shelter, energy, and medicine may exist in abundance, yet remain inaccessible to those who lack the correct credentials, tokens, or permissions.

This inversion marks a critical failure mode. Systems intended to coordinate distribution have come to override the realities they were designed to serve.

### 4.1 Abundance Without Access

In many contemporary contexts, scarcity is no longer primarily material. Grocery stores are stocked, warehouses are full, and supply chains function continuously. Nevertheless, access to these goods depends entirely on authorization mechanisms such as money, accounts, receipts, and databases.

When authorization fails, access is denied, regardless of physical sufficiency. Hunger can exist alongside surplus. Homelessness can coexist with empty buildings. Tools can sit idle while labor remains unused.

In such cases, scarcity is not natural. It is procedural.

### 4.2 Symbols as Primary Reality

Authorization systems operate by privileging symbols over substance. A receipt becomes more real than nourishment. A transaction record outweighs human need. Compliance replaces sufficiency as the criterion for legitimacy.

This inversion produces a condition in which representational abstractions are treated as primary reality, while the physical processes they are meant to coordinate become conditional and subordinate. Human survival itself is rendered secondary to abstract validation, such that access to essential goods depends not on material sufficiency but on symbolic compliance.

Systems organized in this way remain stable only insofar as their representational layer functions without interruption. When that symbolic layer fails, no material or institutional fallback remains, and the systems apparent order rapidly collapses.

### **4.3 Hoarding as a Structural Outcome**

Authorization-based access systems impose no natural upper bound on accumulation. Actors with sufficient authorization may acquire far more resources than they can use, store them indefinitely, or withhold them for speculative purposes.

Conversely, those without authorization are excluded entirely, even when goods are plentiful and waste is evident. This asymmetry is not a moral anomaly; it is a predictable consequence of a system that lacks decay functions, usage constraints, or stewardship requirements.

In historical societies, hoarding of essential goods was often limited by spoilage, social obligation, or direct redistribution during crises. Modern systems remove these constraints while preserving denial.

### **4.4 Fast Control Over Slow Need**

Authorization systems are fast. They are legible, enforceable, and easily optimized. Human needs, by contrast, are slow, contextual, and difficult to quantify.

When fast authorization is allowed to dominate slow need, the system optimizes for procedural correctness rather than human survival. Under stress, such systems do not bend; they break.

### **4.5 Summary**

A civilization that treats authorization as more real than nourishment has inverted its priorities. The resulting system may appear orderly, efficient, and rational, yet it fails its most basic function: sustaining human life.

In the next section, this inversion will be examined in the cognitive domain, where attention itself has become a slow resource subject to fast extraction.

## **5 Attention as a Slow Resource**

Among the slow variables most heavily stressed by modern systems is human attention. Attention is the substrate of learning, judgment, skill acquisition, and social coherence. It develops gradually, requires stability, and degrades under persistent interruption.

Despite this, many contemporary technologies explicitly optimize for rapid engagement, novelty, and reaction speed. This optimization directly conflicts with the conditions under which attention and general intelligence can be maintained.

## **5.1 Attention and Learning**

Learning depends on sustained focus, repetition, and the ability to form long temporal associations. Skills compound over time only when the environment remains sufficiently stable for patterns to be recognized and internalized.

Frequent interruption fragments this process. When attention is repeatedly redirected, cognitive resources are spent on context reconstruction rather than understanding. Over time, this reduces the capacity for deep reasoning and generalization. Attention, once degraded, is slow to recover.

## **5.2 Fast Interfaces and Cognitive Erosion**

Many digital platforms are designed to maximize interaction frequency. Techniques such as infinite scroll, variable reward schedules, algorithmic novelty injection, and constant notifications exploit short-term attentional reflexes.

These mechanisms are effective at increasing engagement metrics, but they do so by consuming the very resource that enables meaningful use of information. The system grows faster by weakening its users. This represents a direct instance of fast processes destroying slow substrates.

## **5.3 Volatility Versus Error**

Errors in tools and systems are often tolerable if behavior is predictable. Stable systems allow users to learn workarounds, develop expertise, and build reliable mental models.

Volatility is more damaging than error. When interfaces change continuously, even small modifications impose cognitive costs. Skills no longer compound. Mastery becomes impossible.

Tools such as long-lived command-line interfaces demonstrate this principle. Their persistence allows users to build durable competence despite imperfections.

## **5.4 Generality and Skill Decay**

Generality refers to the ability to transfer knowledge across contexts. It depends on deep understanding rather than surface familiarity.

Systems optimized for rapid consumption tend to narrow competence. Users become proficient at navigating the interface but less capable of reasoning beyond it. Knowledge becomes siloed, reactive, and brittle.

As generality declines, individuals become increasingly dependent on the very systems that caused the decline, reinforcing the extraction loop.

## 5.5 Summary

Attention is a slow, regenerative resource that cannot be safely optimized for short-term throughput. Technologies that increase engagement by degrading attention reduce human capability over time, regardless of their convenience or scale.

The following section situates this failure mode within a broader historical context, demonstrating that the distinction between fast and slow resources is neither novel nor accidental.

# 6 Historical Recognition of Slow Constraints

The distinction between fast and slow resources is not a modern discovery. Pre-industrial societies, constrained by local feedback and limited transport, often developed explicit mechanisms to manage slow variables such as food, labor, and ecological capacity.

These systems were not idealized or uniformly just, but they frequently embedded safeguards that prevented rapid extraction from overwhelming regenerative processes.

## 6.1 Seasonality and Storage

Agrarian societies organized production around seasonal cycles. Periods of abundance were followed by scarcity, and survival depended on the ability to store resources without exhausting future supply.

Grain storage, preservation techniques, and communal reserves served as buffers against variability. These practices reflected an implicit understanding that immediate consumption must be constrained to ensure long-term viability.

## 6.2 Light and Heavy Resources

Historical statecraft traditions distinguished between resources that were abundant and those that were scarce, often treating the same resource differently depending on temporal conditions.



When a resource was plentiful, it could circulate freely. When scarce, its distribution was regulated to prevent destabilization. This dynamic classification prevented markets or authorities from extracting beyond regenerative limits.

Such distinctions encode a principle absent from many modern systems: scarcity is not static, and governance must adapt accordingly.

### **6.3 Limits on Accumulation**

In many historical contexts, accumulation of essential goods was constrained by spoilage, taxation, or social obligation. Hoarding food during famine was often punished, not because of moral outrage, but because it threatened collective survival.

These constraints acted as natural decay functions, preventing unchecked accumulation and ensuring circulation under stress.

### **6.4 Local Feedback and Accountability**

Pre-modern systems were often characterized by direct feedback. Environmental degradation, food shortages, and social unrest manifested locally and rapidly.

This proximity enforced restraint. Decisions that undermined slow variables produced immediate consequences, discouraging persistent over-extraction.

Modern systems, by contrast, frequently displace consequences across distance and time, weakening corrective feedback.

### **6.5 Summary**

Historical societies recognized, implicitly or explicitly, that fast extraction must be limited by slow regeneration. While technologically constrained, they often preserved resilience through storage, buffering, and accumulation limits.

The erosion of these principles in modern systems has not eliminated scarcity, but displaced it in time and space.

The next section examines how these insights can inform contemporary design, outlining principles that prioritize persistence over efficiency.

## **7 Design Principles for Persistence**

If the failures described thus far are architectural rather than moral, then remedies must also be architectural. The goal is not to optimize existing systems further, but to impose constraints that preserve the slow variables upon which all optimization depends.

This section outlines design principles that prioritize persistence over efficiency.

## 7.1 Constraint Before Optimization

Optimization should occur only within boundaries that protect regenerative substrates. Systems must first define what may not be degraded, regardless of short-term gain.

Such non-negotiable constraints include maintaining ecological regeneration rates above depletion thresholds, limiting extraction to levels that do not compromise future availability, guaranteeing baseline access to essential goods, and enforcing cognitive load limits within information systems. In the absence of such constraints, optimization processes tend to converge on local maxima that deliver short-term gains while exhausting the global capacity on which continued operation depends.

## 7.2 Upper Bounds on Accumulation

Persistent systems require explicit limits on accumulation, particularly with respect to essential resources. In the absence of such limits, unrestricted hoarding produces structural asymmetries that destabilize access and amplify the effects of disruption.

Upper bounds on accumulation may be implemented through mechanisms such as progressive decay of unused stock, increasing costs associated with long-term storage, redistribution triggers activated under conditions of scarcity, or stewardship obligations tied to continued ownership.

These mechanisms are not punitive in nature; rather, they function as stabilizing constraints that preserve circulation and resilience under stress.

## 7.3 Lower Bounds on Access

Persistence requires that access to essentials not fall below survivable thresholds. Food, water, shelter, energy, and basic tools must remain accessible independent of market volatility.

This does not imply uniform distribution, but it does require that authorization systems cannot fully override material sufficiency. Symbolic access must remain subordinate to physical necessity.

## 7.4 Persistence Contracts

Systems should encode explicit persistence contracts that guarantee core behaviors, interfaces, and access pathways will not change arbitrarily or at excessive speed. Such contracts are particularly important for software tools used in long-term skill accumulation, for infrastructure systems upon which daily survival depends, and for institutional procedures

governing access to essential goods and services. Predictability enables learning, and stability enables mastery; without these conditions, competence cannot compound and reliance becomes brittle rather than resilient.

## 7.5 Redundancy and Slack

Efficiency-driven systems often eliminate redundancy and slack as waste. Persistent systems treat them as insurance.

Redundancy allows for failure without collapse. Slack absorbs variability. Together, they prevent cascading breakdowns when assumptions fail.

## 7.6 Local Regeneration Loops

Where possible, production and consumption should be coupled locally. Short feedback loops improve accountability, reduce transport dependency, and preserve contextual knowledge.

Local regeneration does not preclude global exchange, but it prevents total dependence on distant systems whose failure modes are opaque.

## 7.7 Summary

Persistence-first design replaces the question “How fast can this system operate?” with “How long can this system continue?”

Without explicit constraints protecting slow variables, efficiency becomes extractive. With them, optimization can proceed without undermining its own foundation.

The final section synthesizes these principles and restates the central argument.

# 8 Conclusion

The failures examined in this essay do not arise from insufficient optimization, inadequate technology, or individual malice. They arise from a systematic inversion of priorities in which efficiency is pursued without regard for persistence.

When fast variables are allowed to dominate slow ones, systems become brittle. They extract value rapidly while eroding the substrates that make extraction possible. The resulting collapse is often misinterpreted as external shock or moral failure, when it is in fact a predictable outcome of architectural design.

Across domains—logistics, food systems, authorization mechanisms, digital platforms, and cognitive environments—the same pattern recurs. Abstractions are treated as primary realities.

Symbols outrank substance. Speed outranks durability. Throughput outranks learning. Access is mediated by authorization rather than sufficiency.

These systems can function impressively under ideal conditions. Their danger lies in their inability to adapt, recover, or endure. When stress increases, they fail suddenly and disproportionately.

The alternative proposed here is not a rejection of efficiency, markets, or technology. It is a reordering of design priorities. Persistence must come first. Optimization must operate within constraints that preserve ecological regeneration, cognitive capacity, and social continuity.

Civilizations do not collapse because they are insufficiently clever. They collapse because they permit short-term optimization to destroy long-term viability.

The design question is therefore not how to accelerate progress, but how to ensure that progress remains possible.

## **A On the Misclassification of Scarcity**

A recurring failure mode in modern systems is the treatment of scarcity as a static condition rather than a dynamic one. Resources are often classified as either scarce or abundant without regard to temporal variation, regeneration rates, or contextual demand. This misclassification allows systems to behave as though certain resources are permanently available, even when their continued availability depends on slow and fragile processes.

When scarcity is treated as fixed, optimization proceeds as though extraction carries no future cost. In reality, scarcity fluctuates, and governance systems that fail to adapt to these fluctuations amplify instability. Historical practices that distinguished between conditions of abundance and shortage implicitly recognized this dynamic character. Modern systems frequently erase it in favor of uniform rules that privilege speed and legibility over adaptability.

## **B On Symbolic Legibility and System Fragility**

Large-scale systems favor variables that are easily measured, transmitted, and enforced. Symbolic representations such as prices, metrics, credentials, and permissions possess these properties and therefore tend to dominate decision-making. Physical, ecological, and cognitive processes, by contrast, are difficult to quantify and resist central abstraction.

This asymmetry produces fragility. Systems optimize what they can see and ignore what they cannot. Over time, the unseen substrates degrade until symbolic control mechanisms are no longer sufficient to maintain order. At that point, failure appears sudden, despite having been structurally inevitable.

## **C On Volatility as a Hidden Cost**

Instability in system behavior imposes cognitive and operational costs that are rarely captured by efficiency metrics. When interfaces, procedures, or rules change frequently, users are forced to continually reorient, relearn, and compensate. This overhead reduces effective capacity even when nominal functionality improves.

Volatility differs from error in that it prevents adaptation. Errors can be learned and mitigated; volatility cannot. Systems that prioritize rapid iteration without persistence contracts therefore undermine skill accumulation and institutional memory, reducing long-term performance despite short-term gains.

## D On Locality and Feedback

The separation of production from consumption weakens feedback loops that would otherwise constrain extraction. When consequences are displaced geographically or temporally, decision-makers are insulated from the effects of their actions. This insulation enables sustained overuse of slow resources until failure propagates back through the system.

Local regeneration loops shorten feedback cycles and reintroduce accountability. They do not eliminate exchange or specialization, but they prevent total dependence on distant systems whose failure modes are opaque. Persistence requires that at least some essential capacities remain locally recoverable.

## E On Persistence as a Design Criterion

Persistence is not the absence of change, nor is it resistance to innovation. It is the capacity of a system to continue operating while absorbing disturbance and learning over time. Designing for persistence requires prioritizing continuity, regeneration, and recoverability over maximal throughput.

Systems that endure are not those that optimize most aggressively, but those that constrain optimization to preserve their own conditions of possibility. When persistence is treated as a primary design criterion, efficiency becomes a secondary and bounded objective rather than an overriding imperative.

## F Formal Definitions and Terminology

For clarity, this appendix provides precise definitions of key terms used throughout the essay. These definitions are descriptive rather than prescriptive and are intended to reduce ambiguity rather than introduce new theoretical commitments.

A *fast variable* is a system variable that changes rapidly relative to the timescale of system recovery. Fast variables are typically easy to measure, respond quickly to incentives, and are readily abstracted into symbolic representations such as prices, metrics, or signals. Examples include transaction volume, engagement rates, shipping speed, authorization states, and short-term profit.

A *slow variable* is a system variable that changes gradually and requires extended periods of stability to regenerate. Slow variables are often difficult to measure directly, context-dependent, and resistant to abstraction. Examples include ecological fertility, human attention span, skill accumulation, institutional trust, and material sufficiency.

A system exhibits *persistence* if it can continue functioning over long time horizons without degrading the slow variables on which it depends. Persistence is not equivalent to stasis; it includes the capacity to adapt, repair, and learn while preserving core functionality.

An *optimization process* is any mechanism that iteratively adjusts system behavior to maximize or minimize a target variable. Optimization becomes extractive when the target variable is fast and unconstrained by the regeneration limits of relevant slow variables.

## G A Control-Theoretic Perspective on Persistence

From a control-theoretic perspective, many contemporary systems suffer from improperly bounded feedback loops. Controllers are designed to optimize observable outputs while lacking sensors for critical internal state variables. As a result, the system appears stable until hidden state degradation reaches a critical threshold.

In such systems, fast variables function as control signals, while slow variables function as internal state. When control signals are optimized without regard to state depletion, the system exhibits a form of delayed instability. Corrective feedback arrives only after irreversible damage has occurred.

Well-designed control systems incorporate constraints that limit actuation based on internal state. In biological systems, this appears as fatigue, satiety, and regeneration limits. In engineered systems, it appears as safety margins, rate limiters, and fail-safe mechanisms.

Modern socio-technical systems often lack equivalent constraints. Authorization systems enforce correctness of procedure but do not enforce sufficiency of state. Optimization algorithms adjust behavior in response to fast signals while remaining blind to cumulative degradation.

Persistence-first design can therefore be understood as the introduction of state-aware constraints into control architectures. These constraints do not eliminate optimization, but bound it such that control actions cannot drive the system outside recoverable regions of state space.

When persistence is treated as a control objective rather than an external moral consideration, system behavior changes fundamentally. Stability, recoverability, and learning become primary design criteria, and efficiency emerges as a secondary property rather than a dominant goal.

## H Software Systems and Interface Stability

Software systems provide a particularly clear illustration of the tension between efficiency and persistence. Digital tools are often updated rapidly in pursuit of new features, performance

improvements, or engagement metrics. While such changes may offer short-term gains, they frequently impose hidden costs by disrupting learned behavior and invalidating accumulated skill.

Stable interfaces allow users to form durable mental models. When behavior remains predictable over long periods, users adapt to imperfections, develop workarounds, and achieve mastery. In contrast, frequent interface changes require continual relearning, increasing cognitive load and reducing effective competence even when nominal functionality improves.

Backward compatibility serves as a persistence mechanism in software systems. It preserves the validity of prior knowledge and prevents skill decay. When backward compatibility is sacrificed in favor of rapid iteration, systems prioritize short-term novelty over long-term usability. The result is a user population that remains perpetually novice, dependent on documentation and automation rather than understanding.

From a persistence-first perspective, software updates should be constrained by explicit guarantees regarding core behaviors. Change should be additive rather than disruptive, and interface contracts should be treated as long-lived commitments rather than provisional conveniences.

## **I Food Systems, Transport, and Local Regeneration**

Food systems demonstrate how efficiency metrics can obscure physical reality. Global transport networks enable food to be shipped across vast distances at low apparent cost, encouraging specialization and export-oriented production. While this arrangement increases short-term availability, it weakens local resilience and amplifies vulnerability to disruption.

Transport-intensive food systems externalize energy consumption, environmental degradation, and logistical risk. When fuel prices rise or supply chains fail, regions that have lost local production capacity experience sudden scarcity despite favorable ecological conditions.

Persistent food systems prioritize local regeneration loops. Production, processing, and consumption remain coupled closely enough to preserve feedback and accountability. Storage and preservation buffer seasonal variability, reducing dependence on continuous throughput. Global exchange may supplement these systems, but it does not replace local sufficiency.

The persistence of a food system is measured not by volume moved or distance covered, but by its ability to continue feeding a population under stress.

## **J Authorization, Legibility, and Institutional Design**

Institutions rely on legibility to function at scale. Rules, records, and authorization mechanisms allow coordination among large populations. However, when legibility becomes the



primary design objective, institutions risk detaching from the realities they govern.

Authorization systems that operate independently of material sufficiency produce rigid outcomes. Compliance is rewarded even when it undermines purpose, while deviation is punished even when it preserves function. Over time, institutional behavior converges on procedural correctness rather than effectiveness.

Persistence-oriented institutional design maintains a clear hierarchy between symbolic control and physical reality. Authorization mechanisms serve as tools rather than ends, and exceptions are permitted when adherence would cause systemic harm. Such flexibility does not weaken institutions; it preserves their legitimacy under stress.

## **K Mathematical Sketch of Persistence Constraints**

Although the arguments presented here are primarily conceptual, they admit a simple mathematical framing. Let a system be described by a state vector containing both fast variables and slow variables. Optimization processes act to maximize an objective function defined over fast variables.

Persistence constraints restrict allowable actions based on the state of slow variables. When slow variables approach depletion thresholds, optimization is damped or halted regardless of potential gains in the objective function. This prevents trajectories that yield short-term improvement at the cost of irreversible state loss.

In this framing, collapse occurs when optimization proceeds without state-aware constraints, driving slow variables beyond recoverable regions. Persistence-first design corresponds to enforcing invariant sets within which system trajectories must remain.

This formulation emphasizes that persistence is not an external ethical overlay, but an internal requirement for continued operation.

## **L Concluding Remarks on Design Orientation**

The choice between efficiency-first and persistence-first design is not merely technical. It reflects a decision about what a system is for and how long it is expected to exist. Systems optimized solely for immediate performance may succeed briefly, but they do so by consuming the conditions of their own success.

By contrast, systems designed with persistence as a primary constraint accept limits on optimization in exchange for longevity, adaptability, and resilience. Such systems may appear less efficient when measured narrowly, yet they outperform extractive systems over extended horizons.

The central lesson is therefore architectural rather than moral. When slow variables are protected and regeneration is enforced, efficiency becomes sustainable. When they are ignored, efficiency becomes destructive.

## References

- [1] N. Georgescu-Roegen. *The Entropy Law and the Economic Process*. Harvard University Press, 1971.
- [2] D. H. Meadows. *Thinking in Systems: A Primer*. Chelsea Green Publishing, 2008.
- [3] H. T. Odum. *Environment, Power, and Society*. Wiley-Interscience, 1971.
- [4] C. S. Holling. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*, 4:1–23, 1973.
- [5] J. A. Tainter. *The Collapse of Complex Societies*. Cambridge University Press, 1988.
- [6] H. A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 1962.
- [7] W. R. Ashby. *An Introduction to Cybernetics*. Chapman & Hall, 1956.
- [8] I. Illich. *Tools for Conviviality*. Harper & Row, 1973.
- [9] J. C. Scott. *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*. Yale University Press, 1998.
- [10] B. Latour. *We Have Never Been Modern*. Harvard University Press, 1993.
- [11] O. Peters and M. Gell-Mann. Evaluating gambles using dynamics. *Chaos*, 26(2), 2016.
- [12] C. A. Hidalgo. *Why Information Grows: The Evolution of Order, from Atoms to Economies*. Basic Books, 2015.
- [13] J. Norberg. *Progress: Ten Reasons to Look Forward to the Future*. Oneworld Publications, 2016.