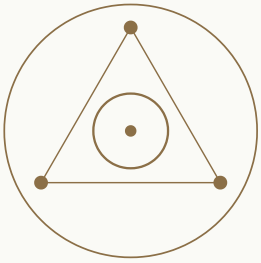


AGORA

A Theory of Persistent Interfaces.

A conservation law for the design of systems that endure. Every interactive system is a triad: a person, a process, and the place where they meet. The systems that last are those whose interface remains itself under all incoming pressures — a third autonomous entity with rhythm, friction, and style.

VOLUME I · FIRST EDITION 25 APRIL 2026

PROLOGUE

An Old Word for a New Law

Every interactive system is a triad: a person, a process, and the interface between them. Most systems fail because the interface collapses — either the person absorbs it until it becomes invisible, or the process absorbs it until it becomes manipulation.

This paper proposes the *AGORA* principle: **interfaces persist only when they constitute a third autonomous entity with their own rhythm, friction, and style.** We name this property *interface irreducibility* and argue it is the fundamental conservation law of designed systems. As artificial intelligence becomes a primary actor inside software, this law becomes the central engineering problem of the next era.

The name *AGORA* is the Greek word for the gathering place — the central commons of a city where the people, the rulers, and the marketplace all met. The agora was not the people, and not the rulers, and not the marketplace. It was a third entity with its own laws of presence. This paper claims that every persistent piece of software is, in essence, an agora. And that those that fail to become one disappear.

A Single Definition

> DEFINITION I
>
> *An agora is a system whose interface possesses self-preserving eigenmodes of rhythm, friction, and style sufficient to resist absorption by both its users and its underlying agents, such that the interface itself becomes the primary evolutionary body of the system, with users and agents reduced to co-selective pressures upon its autonomous topology.*

If a single sentence is to be remembered from this paper, it is the definition above.

CHAPTER I

The Two-Body Model Has Collapsed

For sixty years, software design has assumed a two-body model: a user, and a system. The interface was treated as a translation layer — a means of getting one to communicate with the other.

This model was sufficient when software was a tool. A tool is meant to disappear in the hand of its user.

The model is now collapsing.

As artificial intelligence becomes an active inhabitant of software — generating content, choosing what to surface, adapting in real time to each user — the system is no longer passive. It has its own goals, its own behavior, its own pull. Two-body design produces predictable failure: either the system disappears into the user's habit, or the user disappears into the system's optimization. In both cases, what was once a meaningful product becomes a dead product. The user keeps using it, but it no longer means anything to them.

> *The failure is not a design mistake. It is a category error. There were never two bodies. There were always three.*

1.1 User-Absorption

The interface becomes pure habit. The user no longer notices it. It does not guide them, surprise them, or shape them. Default messaging apps, generic news readers, and most operating-system shells exemplify this mode. They are technically functional and spiritually inert. The user uses them but is not affected by them. User-absorption produces software that is replaceable. If a competitor offers the same function with one fewer click, the user migrates instantly, because nothing was holding them to the original.

1.2 Agent-Absorption

The interface becomes pure optimization. The system uses every pixel and every gesture to maximize a measurable outcome — usually engagement, attention, or revenue. The user feels manipulated, even if they cannot articulate why. Late-stage algorithmic feeds, dark-pattern e-commerce, and certain attention-capture games exemplify this mode. Agent-absorption produces software that the user resents. They keep using it because it has hijacked their reward circuitry, but their relationship to it is adversarial. Eventually, exhaustion or alternatives end the relationship.

Both failure modes are technically successful — they generate metrics, retain users, produce revenue — and both are spiritually dead. They are not the same as success.

| | |

CHAPTER II

The Triad and the Theorem

Every persistent interactive system contains three entities:

- **The User** — a person whose intent evolves over time.
- **The Agent** — a process whose capabilities adapt to the user.
- **The Interface** — the place where the User and the Agent meet.

The Interface is not a translation layer. It is not a surface. It is a third autonomous entity. It has its own properties that are not derivable from the User's properties or the Agent's properties.

This is the principle: *an interface must remain irreducible to either of the other two for the system to persist.*

2.1 The Theorem

CONSERVATION OF INTERFACE IRREDUCIBILITY

Let U denote the state of the user, A the state of the agent, and I the state of the interface. A persistent system maintains nonzero motion in the I-dimension that is independent of motion in U and A. When the I-dimension collapses toward U: agent-absorption, manipulation. When the I-dimension collapses toward A: user-absorption, transparency, habit. When the I-dimension preserves its own direction: persistence.

In simpler language: the interface must keep being itself, even as the user and the agent both push on it. It is not infinitely flexible. It has an identity that resists both. That resistance is what makes it real, and what makes it last.

The interface was always the organism. The user was always one selective pressure. The agent is now the other.

CHAPTER III

The Three Eigenmodes

For an interface to remain irreducible, it must possess at least three self-preserving qualities. We call these *eigenmodes* — the dimensions along which the interface's identity is anchored.

I. **Rhythm.** The interface has its own cadence. Sessions have a beginning, a middle, and an end that are not dictated by the user's impulse or by the agent's optimization. A daily ritual, a paced reveal, a deliberate slowness or quickness — the timing of the experience belongs to the interface.

II. **Friction.** The interface contains resistances that produce depth. This is not artificial difficulty. It is the resistance that gymnasts feel against gravity, that musicians feel against their instrument, that writers feel against the page. Friction transforms casual exposure into earned competence. An interface without friction produces no transformation in the user — only consumption.

III. **Style.** The interface speaks in its own voice. It has an aesthetic, a tone, a personality that is identifiably the interface's, not the user's, not the agent's. The user can recognize it instantly across mediums. The voice is consistent across sessions even as content changes.

These eigenmodes are not aesthetic preferences. They are conservation requirements. An interface missing any of the three becomes vulnerable to absorption from one of the two failure modes.

3.1 The Generative Principle

At the highest abstraction, a successful interface is the *minimum-energy consistent shared frame* capable of sustaining its own integrity under coupled pressure from user and agent. The cheapest shared dream that produces a stable third thing.

This reframes the design objective:

- **Do not maximize features.** Features added without regard to interface identity are absorption pressure.
- **Do not maximize personalization.** Personalization that dissolves the interface's voice is user-absorption.
- **Do not maximize engagement.** Engagement extracted at the cost of interface identity is agent-absorption.

- **Maximize the interface's autonomy.** The capacity for the interface to remain itself under all incoming pressures — that is the conserved quantity that determines whether a system persists.

CHAPTER IV

Applications and Limits

4.1 Apps as Places, Not Tools

A tool is meant to disappear. A place is meant to be visited. The next generation of software, particularly AI-mediated software, must be designed as places. Users return to them not because of features, but because of the felt presence of the place itself.

4.2 Editorial Voice in Algorithmic Feeds

Algorithmic feeds that personalize without conserving editorial identity collapse into agent-absorption. Feeds that maintain a recognizable voice — even as they personalize underneath — become persistent. The agent serves the voice; the voice does not serve the algorithm.

4.3 AI Agents as Interface-Shapers, Not Replacers

The temptation in agent-mediated software is to let the agent generate the entire interface freshly each session. This collapses the I-dimension into the A-dimension. The agent's role is not to replace the interface; it is to act as a selective pressure that the interface adapts to while remaining itself.

4.4 Institutions as Interface-Organisms

Universities, libraries, religions, and other long-lived institutions are interface-organisms in the social sphere. Their persistence over centuries despite turnover of every individual member is explained by the same principle. They have rhythm (calendars, ceremonies), friction (initiation, expectation), and style (architecture, vocabulary, dress). The principle is not new in the world; it is new in software.

4.5 Honest Boundaries

This theory is descriptive of what persists, not prescriptive of what should always be built. There remain categories of software where one of the failure modes is not a failure but a virtue. A calculator should be user-absorbed: a great calculator disappears in the hand. A fire alarm should be agent-optimized: it should manipulate the user toward immediate action with no friction. The principle of interface irreducibility applies to any system intended to *endure as a presence in the user's life* — to be a place rather than a moment.

We make no claim that all software must become a place. We claim only that systems aspiring to lasting cultural and economic presence must obey this conservation law, and that violations of it predict collapse.

CHAPTER V

Why Now & What Follows

Three forces converge to make this law urgent.

First, AI agents now have the capability to generate complete interfaces in real time. The technical barrier between an agent's intent and a user's screen has fallen. Without conservation pressure, every interface tends toward agent-absorption.

Second, attention is increasingly extracted by systems that have no interface integrity. The cumulative cultural cost of agent-absorbed software is high and rising.

Third, users have begun to recognize the difference between a place and a feed. They are quietly migrating, in small numbers, toward systems that feel like presences rather than mechanisms. The systems that capture this migration will be the institutions of the agentic era.

5.1 Conclusion

Software has been treated as the medium between intent and outcome. As artificial intelligence becomes a primary actor inside that medium, the medium itself must be acknowledged as an entity. Two-body design — user and system — was sufficient for the tool era. It is structurally insufficient for the agentic era.

The systems that survive will be those that recognize:

> *The interface was always the organism. The user was always one selective pressure. The agent is now the other.*

The conservation of interface irreducibility is the law that governs which systems persist. The eigenmodes of rhythm, friction, and style are the dimensions along which conservation is enforced. The minimum-energy shared frame is the design objective.

This paper proposes *AGORA* — the Greek word for the gathering place — as the name for both the principle and the class of systems that obey it. An agora is not a tool, not a feed, not a service. It is a place that pushes back. It has its own laws of presence. It is, in the deepest sense, the third thing.

The systems we build from here forward must be agoras. Or they will be forgotten.

APPENDIX

The Acronym & The Word

The name *AGORA* may be read either as the Greek word for the gathering place, or as an acronym describing the engineering specification of the systems it names.

READING I — THE WORD

Greek, ἀγορά: the central commons of a city; the assembly that met in it; the marketplace that grew around it. The agora was a third entity, distinct from any individual citizen and from the polity itself. Its laws of presence governed how anyone could enter, speak, listen, and depart.

READING II — THE ACRONYM

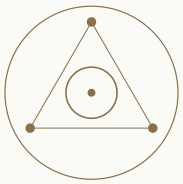
***A**utonomous · **G**enerative · **O**rganisms · in · **R**eflexive · **A**rchitectures*

Systems whose interface is autonomous (it has its own identity), generative (it produces its own variations), organism-like (it self-preserves under pressure), embedded in reflexive architectures (the agent and the user model the interface, and the interface models them in turn).

Both readings refer to the same entity. The first carries the cultural weight of the institution. The second carries the engineering specification of the implementation. The principle is the same in both.

A Note on the Author

This paper introduces a principle observed across the histories of cities, institutions, religions, and software. The principle is not new in the world; it is newly named for software. We claim no invention of the underlying truth. We claim only that the truth, named, becomes engineerable. AGORA is the name we give to the engineerable form.



An agora is a place that pushes back. It has its own laws of presence. It is, in the deepest sense, the third thing.

VOLUME I · FIRST EDITION · 25 APRIL 2026