The Deliberate Collapse of Cognitive Multiplicity: Event-Driven Intelligence, Spherepop, and the Reckless Design of Big Tech Interfaces

Flyxion

December 2025

1 Introduction

The general-purpose computer was not conceived as a content delivery channel, nor as a behavioral conduit optimized for engagement. It was designed as a universal symbolic workspace: a system capable of hosting multiple concurrent processes, representations, and transformations under user control. Early computing environments embodied this commitment explicitly through windows, editable text, inspectable structure, and unrestricted recomposition.

These affordances were not incidental. They instantiated a theory of cognition according to which intelligence is inherently plural, interruptible, and historical. Thought does not occur as a single stream, but as a structured accumulation of events whose ordering, branching, and recombination determine meaning.

This essay argues that contemporary Big Tech platforms have systematically violated this principle. By collapsing multifunctional computing into single-threaded attention systems, modern interfaces erase event-history structure and replace cognitive agency with behavioral containment. Grounded in event-driven cognition and formalized through the Spherepop calculus, this critique frames such design practices as reckless interventions into humanityâĂŹs shared cognitive infrastructure.

2 Event-Driven Cognition and the Primacy of History

Definition 1 (Event)**.** *An event* E *is an irreversible transformation that contributes to the construction history of a cognitive system.*

Definition 2 (Event History)**.** *A history* $\mathcal{H}$ *is a partially ordered sequence of events whose structure determines semantic and functional identity.*

Event-driven cognition holds that meaning, competence, and understanding are not functions of instantaneous system state, but of the trajectory by which that state was reached. Two systems with identical present configurations may differ cognitively if their event histories differ.

This principle applies equally to biological minds and computational tools. Learning is path-dependent. Insight arises from interruption and recombination. Memory is not storage but structured replay. Any tool that supports cognition must therefore preserve access to event histories and allow users to manipulate them explicitly.

General-purpose computers once did precisely this. Copy-and-paste preserved fragments of event history across contexts. Multiple windows externalized parallel histories. Editable text allowed revision trajectories to remain legible. These were not conveniences; they were cognitive necessities.

3 Constraint as the Medium of Intelligence

Definition 3 (Constraint)**.** *A constraint $\mathcal{C}$ is a rule or boundary that restricts the space of admissible event continuations.*

Intelligence does not consist in the absence of constraints, but in the ability to navigate, negotiate, and reconfigure them. Productive constraint sharpens cognition by preserving structure while allowing transformation. Destructive constraint collapses possibility space and suppresses agency.

Event-driven systems require constraints that are transparent, interruptible, and revisable. When constraints are hidden, rigid, or imposed without user control, they destroy the informational content of event histories by preventing branching and recomposition.

Modern platform interfaces increasingly employ constraints not to support cognition, but to stabilize behavior.

4 Single-Threaded Interfaces as Event Suppression

Contemporary applications are increasingly designed around a single dominant attention thread: one feed, one viewport, one sanctioned interaction path at a time. Multitasking is discouraged or technically obstructed. Text selection is restricted. Structure is hidden. Layout is immutable.

These design choices systematically suppress event branching. They collapse parallel histories into a single forced trajectory and prevent users from externalizing their own cognitive structure.

Proposition 1. *Any interface that enforces a single dominant event continuation while suppressing interruption and recomposition necessarily degrades cognitive agency.*

Proof. Cognitive agency requires the ability to redirect event trajectories, preserve alternative branches, and recombine histories. A single enforced continuation eliminates these possibilities, collapsing history into reaction. Therefore agency is reduced. □

The result is not merely inconvenience, but a transformation of cognition itself. Users adapt to the available affordances, internalizing single-threaded habits of attention and expression.

5 Spherepop: Computation as Event Composition

Spherepop is a calculus and operating paradigm explicitly designed around event-driven cognition. Rather than treating computation as state transition, Spherepop models systems as evolving collections of event histories subject to merge, collapse, and constraint operations.

In Spherepop, summarizing and extending are not distinct activities. Both are transformations of event structure: either compressing histories by identifying invariants, or expanding them by reintroducing suppressed branches. Meaning is preserved not by freezing state, but by respecting construction history.

From this perspective, interfaces that prohibit recomposition, inspection, or symbolic manipulation are not simplified systems. They are broken cognitive environments. They prevent users from performing the fundamental operations required to think with machines.

6 The Flattening of Expression on Social Platforms

Social platforms provide a clear example of event suppression through interface design. Early web systems allowed users to express hierarchy, emphasis, and structure directly through markup. These capabilities externalized abstraction and supported complex event histories of thought.

The deliberate removal of typographic control, structural markup, and inspectable representation enforces expressive homogeneity. All utterances are rendered equivalent at the interface level, regardless of internal structure.

Over time, this collapses argument into slogan and reasoning into reaction. The loss is cumulative and civilizational. Describing this as a crime against humanity is not a legal claim, but a moral diagnosis: it names the scale of harm inflicted by the systematic erosion of expressive capacity.

7 Samsung, Android, and the Criminalization of Tinkering

SamsungâĂŹs control over the Android interface provides a particularly stark case of constraint misuse. While Android presents itself as an open platform, Samsung restricts system-level customizationâĂŤsuch as font controlâĂŤto a small, tightly regulated class of approved theme developers.

Access to these capabilities is rationed, monetized, and effectively sold. Ordinary users are prohibited from modifying foundational representational layers of their own devices.

This decision has profound developmental consequences. Historically, the ability to alter fonts, layouts, and system behavior served as an entry point into hacking, cryptography, and systems thinking. By sealing these layers, Samsung suppresses exploratory event histories before they can form.

Remark 1. *This is not a loss of aesthetics, but of cognitive apprenticeship. A generation denied the right to tinker is a generation denied the means to understand.*

It is plausible that such restrictions have delayed the emergence of a broadly technically literate public by a decade or more.

8 Recklessness at Civilizational Scale

These design choices are not isolated mistakes. They constitute a coherent strategy optimized for predictability, monetization, and behavioral control. Short-term engagement is maximized by collapsing event space, while long-term cognitive costs are externalized onto education, politics, and mental health.

This is recklessness in the literal sense: action taken without regard for cumulative consequence. When deployed at planetary scale, such recklessness reshapes humanityâĂŹs cognitive environment.

9 Conclusion: The Right to History

Computers are cognitive tools because they preserve and manipulate event histories. When interfaces respect this fact, they amplify intelligence. When they suppress it, they deform thought.

The fundamental right at stake is the right to history: the right to interrupt, to branch, to recombine, and to tinker. Any system that denies this right in the name of simplicity or profit is not merely badly designed. It is ethically compromised.

If intelligence is to survive the platform era, computing must be reclaimed as a space of event-driven agency rather than behavioral containment. Spherepop is one attempt to formalize this reclamation. Whether or not it succeeds, the principle it embodies is unavoidable: cognition only exists where history is allowed to matter.

A Formal Operational Semantics of Spherepop

This appendix provides a minimal operational semantics for Spherepop, sufficient to make precise the claims in the main text concerning event-driven cognition, recomposition, and constraint-respecting computation. The purpose is not to exhaustively formalize the system, but to demonstrate that the underlying commitments admit a rigorous semantics.

A.1 Event Structures

Definition 4 (Event Token)**.** *An event token $e \in \mathsf{E}$ is an atomic, irreversible operation that transforms a semantic substrate.*

Event tokens are not states. They do not describe configurations but transitions. Once introduced, an event token cannot be removed from history, only summarized or constrained.

Definition 5 (Event History)**.** *An event history $\mathcal{H}$ is a finite partially ordered multiset $(E, \preceq)$, where $E \subset \mathsf{E}$ and $\preceq$ encodes causal precedence.*

Partial order is essential. Independent events may commute, while dependent events preserve order. This allows concurrency without collapsing history into a total sequence.

A.2 Configurations

Definition 6 (Spherepop Configuration)**.** *A Spherepop configuration is a triple*

$$\langle \mathcal{H}, \mathcal{C}, \mathcal{M} \rangle$$

where $\mathcal{H}$ is an event history, $\mathcal{C}$ is a constraint set governing admissible continuations, and $\mathcal{M}$ is auxiliary metadata not semantically decisive.

Only $\mathcal{H}$ participates in semantic identity. Constraints restrict future evolution. Metadata may affect presentation but never alters meaning.

A.3 Core Reduction Rules

Spherepop evolution is defined by small-step transitions over configurations.

Definition 7 (Event Extension)**.**

$$\langle \mathcal{H}, \mathcal{C}, \mathcal{M} \rangle \;\rightarrow\; \langle \mathcal{H} \cup \{e\}, \mathcal{C}, \mathcal{M} \rangle \quad \textit{if } e \textit{ is admissible under } \mathcal{C}$$

This rule introduces a new irreversible event, extending history without collapsing existing structure.

Definition 8 (Branch)**.**

$$\mathcal{H} \;\Rightarrow\; \mathcal{H}_1 \parallel \mathcal{H}_2$$

Branching produces parallel continuations sharing a common prefix. Branching is not copying state; it is duplicating future possibility while preserving shared history.

Definition 9 (Merge)**.**

$$\mathcal{H}_1 \oplus \mathcal{H}_2 \;\rightarrow\; \mathcal{H}_3$$

where $\mathcal{H}_3$ preserves all non-contradictory events and introduces explicit resolution events for conflicts.

Merge is a semantic operation, not a overwrite. Conflicts are recorded as events, not erased.

Definition 10 (Collapse)**.**

$$\text{collapse}(\mathcal{H}) = \mathcal{H}'$$

where $\mathcal{H}'$ replaces a subhistory with an invariant summary event.

Collapse is lossy but explicit. Information loss is represented as an event, preserving epistemic honesty.

A.4 Constraint Dynamics

Definition 11 (Constraint Application)**.**

$$\mathcal{C} \vdash e \quad \textit{or} \quad \mathcal{C} \nvdash e$$

Constraints determine admissibility of future events but do not retroactively modify history. This enforces temporal asymmetry and prevents revisionist semantics.

Proposition 2 (Constraint Monotonicity)**.** *Constraints may restrict future event extensions but cannot remove or alter past events.*

Proof. By construction, constraints are predicates on admissible continuations. No rule permits retroactive modification of $\mathcal{H}$. Therefore past events are invariant. □

This property distinguishes Spherepop from rollback-based state machines and aligns it with irreversible cognition.

A.5 Replay and Equivalence

Definition 12 (Replay)**.** *Replay is the reconstruction of semantic outcomes by reapplying events in $\mathcal{H}$ respecting $\preceq$.*

Theorem 1 (Replay Equivalence)**.** *Any two configurations with isomorphic event histories are semantically equivalent, regardless of metadata or presentation.*

Proof. Semantic outcomes are determined solely by event structure and order. Metadata does not participate in reduction rules. Therefore isomorphic histories replay identically. □

This theorem formalizes the claim that meaning resides in construction history, not surface form.

A.6 Interface Pathologies as Semantic Failures

Interfaces that prohibit branching, merging, inspection, or replay effectively restrict the operational semantics to a degenerate fragment. In such systems:

$$|\text{admissible continuations}| = 1$$

This collapses event-driven computation into a forced linear reaction chain.

Remark 2. *A single-threaded interface is equivalent to imposing a global constraint that forbids all but one future event at each step.*

From a Spherepop perspective, such systems are not merely simplified. They are semantically impoverished, incapable of supporting genuine cognition.

A.7 Relation to Tinkering and Literacy

Tinkering corresponds to low-cost event extension and branching at foundational layers of representation. When systems restrict these operationsâĂŤsuch as prohibiting font manipulation or symbolic inspectionâĂŤthey block entire regions of event space.

The resulting loss is not recoverable through higher-level abstractions. If early event histories cannot form, later competence cannot emerge.

A.8 Summary

Spherepop operational semantics formalize a simple principle: intelligence requires history, and history requires freedom to branch, merge, and recombine events. Any computing system that suppresses these operations does not merely limit functionality; it suppresses the conditions under which thinking itself is possible.

B Spherepop Core Syntax (BNF Grammar)

This appendix specifies a minimal concrete syntax for Spherepop sufficient to express event construction, branching, merging, collapse, and constraint application. The grammar is intentionally small, reflecting SpherepopâĂŹs role as an event calculus rather than a general-purpose programming language.

B.1 Lexical Elements

Identifiers range over event names, constraint labels, and metadata keys.

$$\langle id \rangle ::= [a\text{-}zA\text{-}Z][a\text{-}zA\text{-}Z0\text{-}9\_]^*$$

B.2 Programs

$$\langle program \rangle ::= \langle stmt \rangle^*$$

A program is a sequence of statements interpreted as successive event-history transformations.

B.3 Statements

$$\begin{aligned}\langle stmt \rangle ::= & \texttt{event}\ \langle id \rangle \\ & \mid \texttt{branch}\ \langle block \rangle\ \texttt{||}\ \langle block \rangle \\ & \mid \texttt{merge}\ \langle id \rangle\ \langle id \rangle \\ & \mid \texttt{collapse}\ \langle id \rangle \\ & \mid \texttt{constrain}\ \langle constraint \rangle \\ & \mid \texttt{meta}\ \langle id \rangle\ =\ \langle value \rangle\end{aligned}$$

B.4 Blocks

$$\langle block \rangle ::= \texttt{\{}\ \langle stmt \rangle^*\ \texttt{\}}$$

Blocks define local continuations sharing a common history prefix.

B.5 Constraints

$$\langle constraint \rangle ::= \texttt{allow}\ \langle id \rangle \mid \texttt{deny}\ \langle id \rangle \mid \texttt{require}\ \langle id \rangle$$

Constraints are predicates on admissible future events. They do not modify past history.

B.6 Values

$$\langle value \rangle ::= \langle id \rangle \mid \langle string \rangle \mid \langle number \rangle$$

Metadata values affect presentation or annotation only and never participate in semantic reduction.

B.7 Design Rationale

The grammar excludes assignment, mutation, and control flow constructs typical of state-based languages. All expressive power arises from event introduction, branching, recomposition, and constraint modulation. This enforces the event-history ontology at the syntactic level.

C Typed Spherepop: A Minimal Type System

This appendix introduces a typed variant of Spherepop designed to enforce semantic invariants without collapsing event-driven flexibility. Types classify events and histories, not machine states.

C.1 Types

Definition 13 (Core Types)**.**

$$\begin{aligned} \tau ::= & \texttt{Event} \\ & \mid \texttt{History} \\ & \mid \texttt{Constraint} \\ & \mid \texttt{Meta} \\ & \mid \tau \to \tau \end{aligned}$$

Events and histories are distinct types. No coercion exists from history to event or vice versa.

C.2 Typing Contexts

A typing context Γ maps identifiers to types.

$$\Gamma ::= \emptyset \mid \Gamma, x : \tau$$

C.3 Typing Judgments

Typing judgments have the form

$$\Gamma \vdash s : \tau$$

indicating that statement s produces a semantic object of type τ.

C.4 Typing Rules

Theorem 2 (Event Introduction)**.**

$$\frac{}{\Gamma \vdash \mathit{event}\, e : \mathit{Event}}$$

Theorem 3 (History Extension)**.**

$$\frac{\Gamma \vdash h : \mathit{History} \quad \Gamma \vdash e : \mathit{Event}}{\Gamma \vdash h + e : \mathit{History}}$$

Event addition extends history but never alters existing structure.

Theorem 4 (Branching)**.**

$$\frac{\Gamma \vdash h : \mathit{History}}{\Gamma \vdash \mathit{branch}(h) : \mathit{History} \times \mathit{History}}$$

Branching duplicates future possibility while preserving shared past.

Theorem 5 (Merge)**.**

$$\frac{\Gamma \vdash h_1 : \mathit{History} \quad \Gamma \vdash h_2 : \mathit{History}}{\Gamma \vdash \mathit{merge}(h_1, h_2) : \mathit{History}}$$

Merge produces a new history containing explicit resolution events.

Theorem 6 (Collapse)**.**

$$\frac{\Gamma \vdash h : \mathit{History}}{\Gamma \vdash \mathit{collapse}(h) : \mathit{History}}$$

Collapse is type-preserving but informationally lossy.

Theorem 7 (Constraint Application)**.**

$$\frac{\Gamma \vdash c : \mathit{Constraint} \quad \Gamma \vdash h : \mathit{History}}{\Gamma \vdash c(h) : \mathit{History}}$$

Constraints restrict admissible future extensions without modifying past events.

C.5 Type Safety

Theorem 8 (History Preservation)**.** *Well-typed Spherepop programs cannot delete or reorder past events.*

Proof. No typing rule permits removal or mutation of an existing history. All constructors either extend, branch, merge, or summarize histories while preserving causal order. □

This theorem formalizes the ethical and cognitive commitment of Spherepop: history, once created, is inviolable.

C.6 Typed Interfaces and Cognitive Guarantees

Interfaces that prohibit branching, merging, or collapse operations correspond to type-erasing projections of Spherepop. Such projections are not type-safe with respect to cognitive agency: they eliminate entire classes of well-typed programs.

From this perspective, restrictive platforms are not merely limited implementations. They are unsound semantic environments incapable of expressing valid event-driven computations.

C.7 Summary

The typed Spherepop calculus demonstrates that event-driven cognition admits formal guarantees without reverting to state-based control. By typing histories rather than configurations, Spherepop enforces semantic honesty while preserving the freedom required for thinking.

D Correspondence Theorem: Event Histories and Versioned Computation

This appendix establishes a formal correspondence between Spherepop event-driven semantics and a broad class of real-world computational systems, including version control systems, event-sourced databases, and cognitive replay models. The theorem clarifies which properties are preserved under correspondence and which failures arise when interfaces collapse event structure.

D.1 Reference Systems

We consider three classes of systems:

Definition 14 (Event-Sourced System)**.** *An event-sourced system is one in which the authoritative record is an append-only log of events, and current outcomes are derived by replay.*

Definition 15 (Versioned System)**.** *A versioned system is one in which histories may branch, merge, and summarize, producing a directed acyclic graph of revisions.*

Definition 16 (State-Centric System)**.** *A state-centric system is one in which only the current configuration is retained, and prior transitions are discarded or made inaccessible.*

Spherepop belongs to the first two classes and explicitly rejects the third.

D.2 Correspondence Mapping

Definition 17 (Correspondence Mapping)**.** *A correspondence mapping $\mathcal{F}$ from a Spherepop configuration*

$$\langle \mathcal{H}, \mathcal{C}, \mathcal{M} \rangle$$

to a reference system is a structure-preserving map such that:

1. *Events map to atomic log entries or commits;*

2. *Partial order maps to causal or dependency order;*

3. *Branch corresponds to divergent histories;*

4. *Merge corresponds to explicit reconciliation operations;*

5. *Collapse corresponds to squashing or summarization.*

The mapping need not be bijective, but must preserve replay semantics.

D.3 Main Correspondence Theorem

Theorem 9 (Event-History Correspondence)**.** *For any Spherepop history $\mathcal{H}$, there exists an event-sourced or versioned system S and a correspondence mapping $\mathcal{F}$ such that replay of $\mathcal{H}$ and replay of $\mathcal{F}(\mathcal{H})$ produce semantically equivalent outcomes.*

Proof. Spherepop histories are finite partially ordered sets of irreversible events. Event-sourced and versioned systems are defined by the same structural properties: append-only logs with explicit branching and merging. Mapping each Spherepop event to a log entry preserves order and causality. Replay equivalence follows from identical dependency structure. □

This establishes that Spherepop semantics are not exotic: they formalize practices already relied upon in reliable computing systems.

D.4 Failure of Correspondence for Single-Threaded Interfaces

We now state the critical negative result.

Theorem 10 (No Correspondence for Collapsed Interfaces)**.** *There exists no correspondence mapping from a Spherepop history with branching or merge structure to a strictly single-threaded, state-centric interface that preserves semantic replay.*

Proof. Single-threaded state-centric interfaces discard all but the most recent configuration and prohibit branching. Any mapping from a branching history to such a system must either erase alternative paths or linearize them without record. In either case, replay equivalence fails, since distinct histories map to indistinguishable states. □

This is not a limitation of implementation, but a structural impossibility.

D.5 Corollary: Cognitive Degradation

Any interface whose operational semantics correspond to a state-centric, single-threaded system cannot faithfully support event-driven cognition.

Proof. Event-driven cognition depends on access to alternative histories, interruptions, and recomposition. By the previous theorem, such structures cannot be represented or replayed in single-threaded interfaces. Therefore cognitive semantics are necessarily degraded. □

This corollary formalizes the main argument of the essay: interface simplification that removes history is not neutral abstraction but semantic destruction.

D.6 Interpretation

The correspondence theorem shows that Spherepop aligns with the most robust paradigms of reliable computationâĂŤthose that preserve history and support replay. The systems that dominate contemporary consumer software, by contrast, correspond to the weakest possible semantic class.

This gap explains why such systems scale economically yet fail cognitively. They are optimized for control, not for understanding.

D.7 Summary

Spherepop is not a speculative alternative to existing computation, but a formal distillation of what already works where correctness, accountability, and learning matter. The refusal of Big Tech interfaces to support these semantics is therefore not an engineering necessity, but a choiceâĂŤone whose cognitive consequences are now unavoidable.