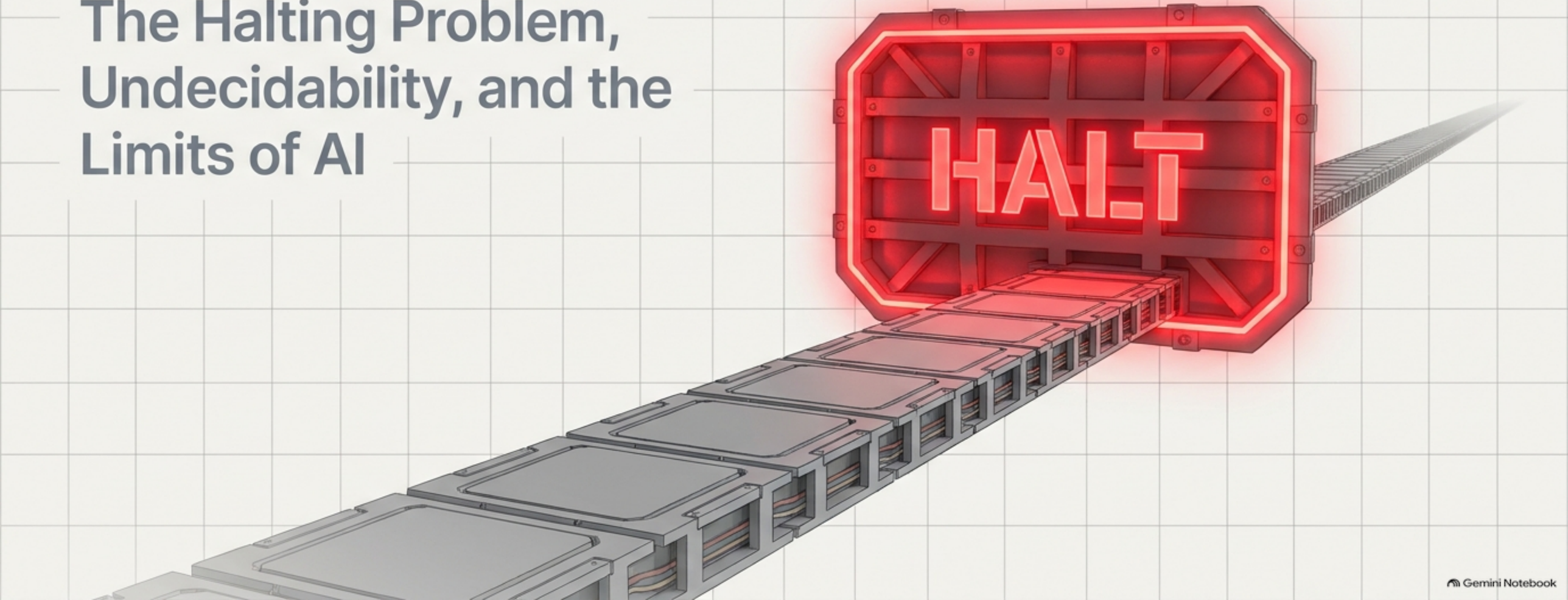
Hard Problems in Computing Computing — S08E12

The Halting Problem, Undecidability, and the Limits of AI

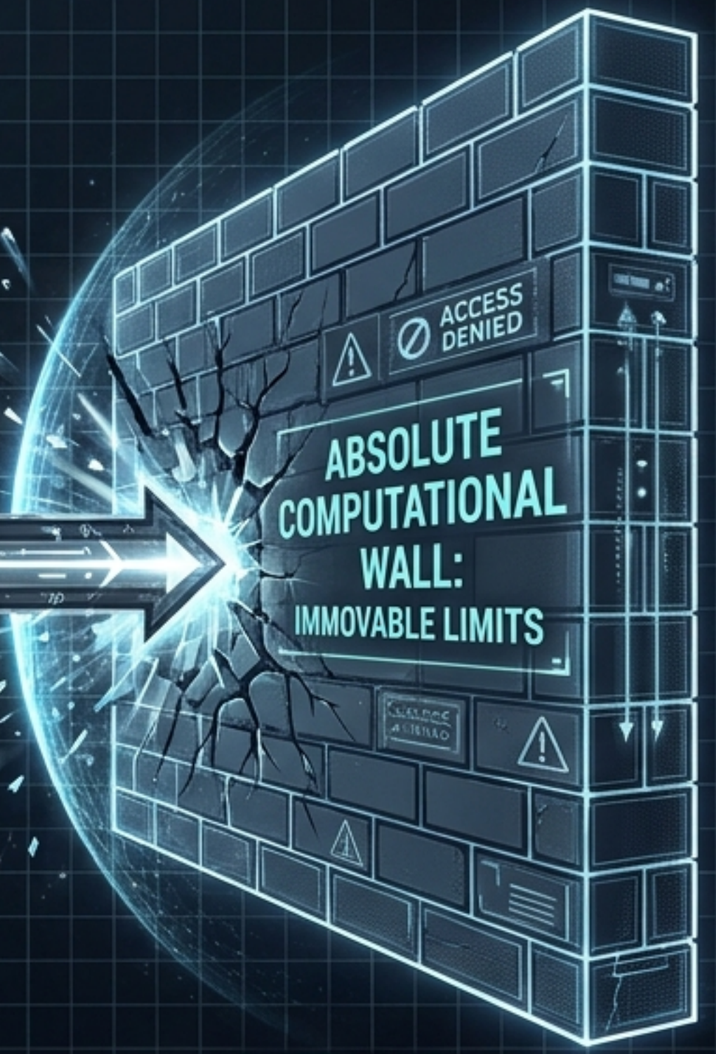


Gemini Notebook

While S08E11 showed us how Zero-Knowledge Proofs manipulate mathematical probability to create cryptographic miracles, today we hit the absolute, unyielding wall of computation.



CRYPTOGRAPHIC VICTORY:
PROBABILITY MANIPULATION



SYSTEM HALT:
UNYIELDING COMPUTATION.

ALARM STATE: CRITICAL | COMPUTATIONAL BOUNDARY BREACH ATTEMPT FAILED

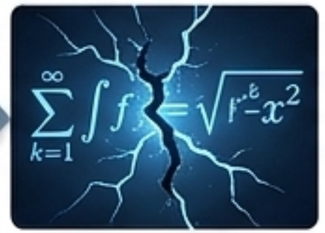
Gemini Notebook



The Dream: Hilbert's Entscheidungsproblem



The Math: Cantor's Cardinality Gap



The Crack: Gödel's 1931 Incompleteness



The Limit: The Church-Turing Thesis

The Universal Limit.

Why classical CPUs, modern compilers, and frontier LLMs all share the exact same computability limits.

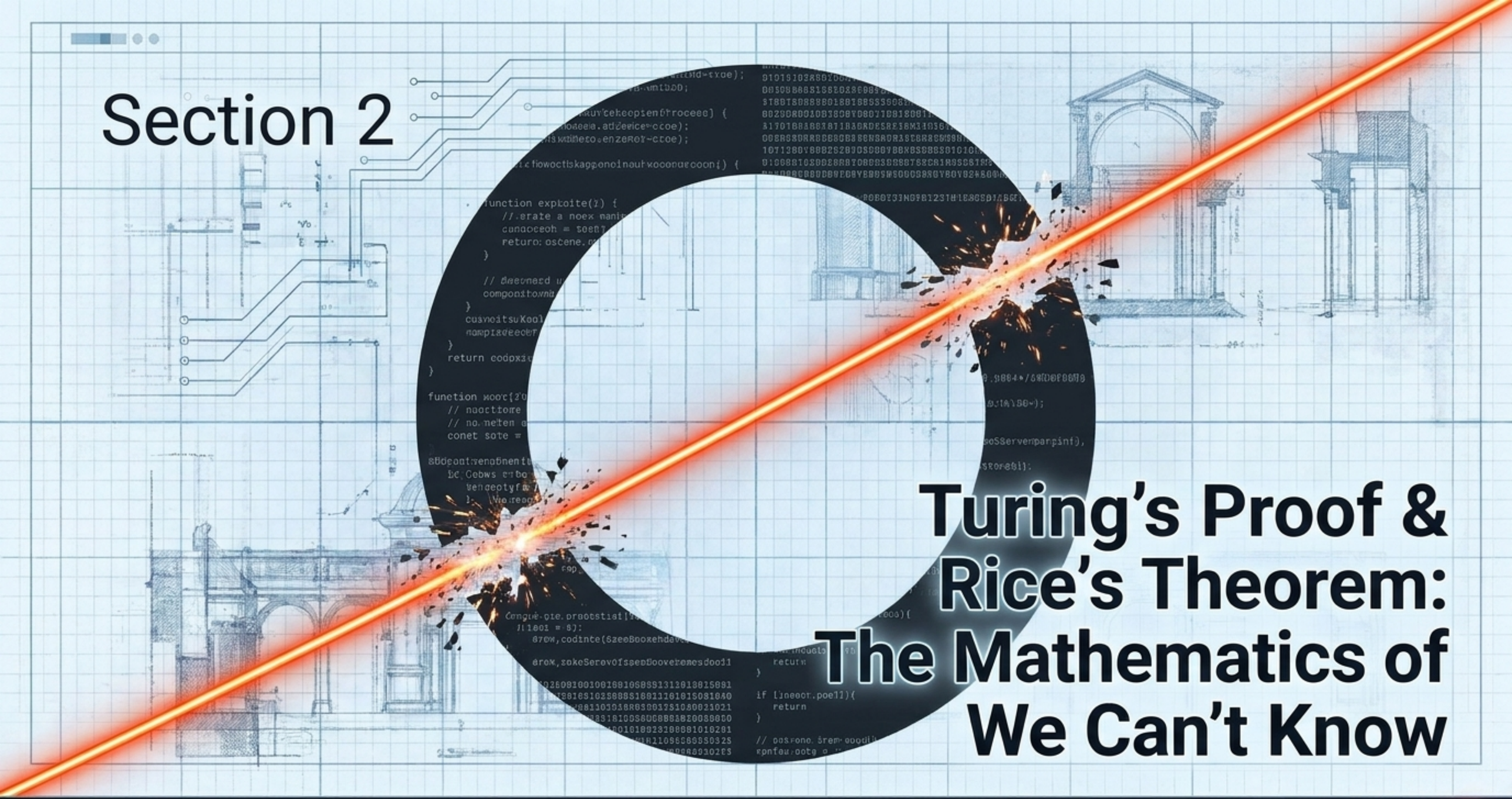
No amount of compute can bypass the Halting Problem.

ALARM STATE: CRITICAL | COMPUTATIONAL BOUNDARY BREACH ATTEMPT

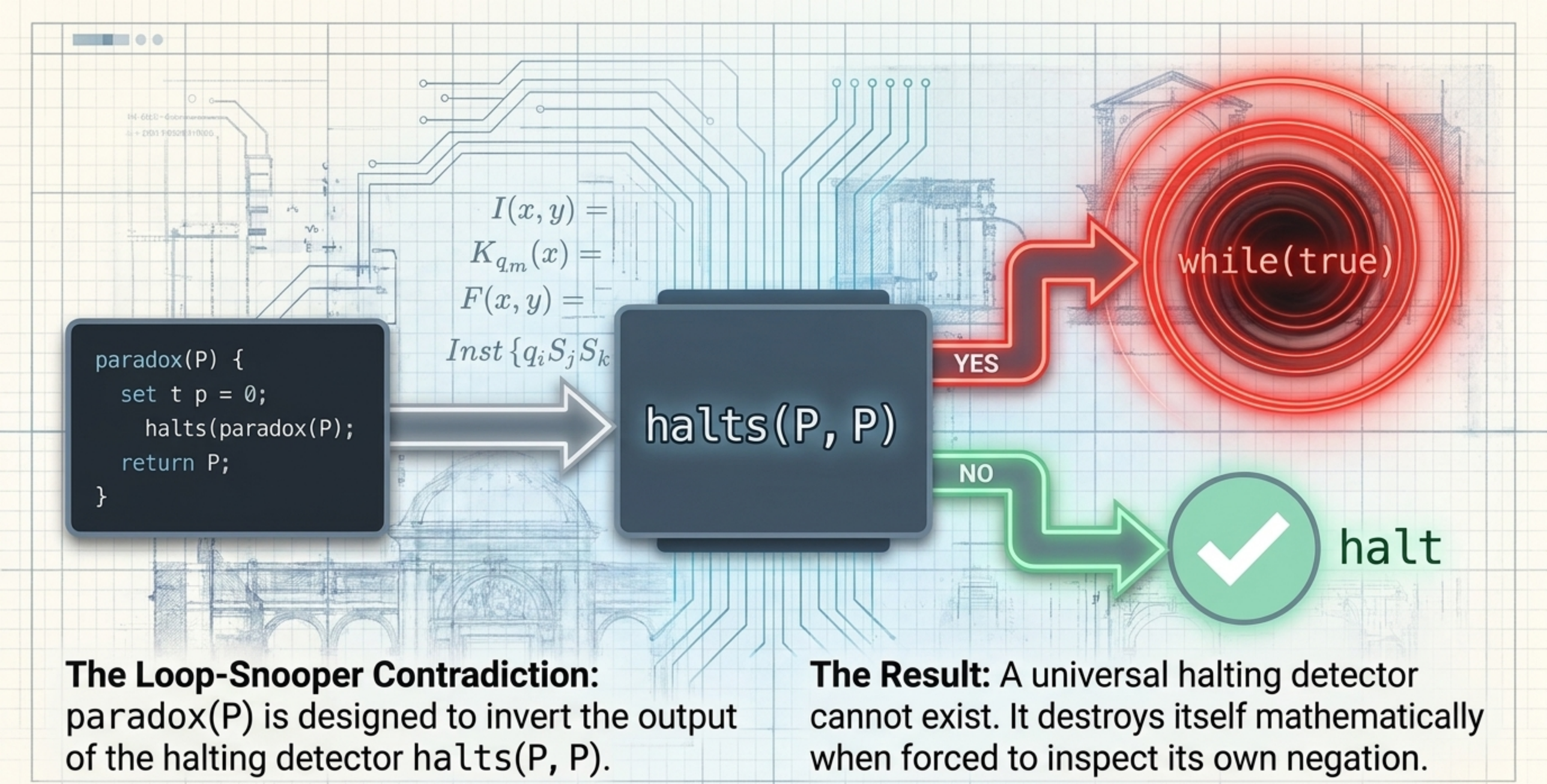
Gemini Notebook

Section 2

Turing’s Proof & Rice’s Theorem: The Mathematics of We Can’t Know



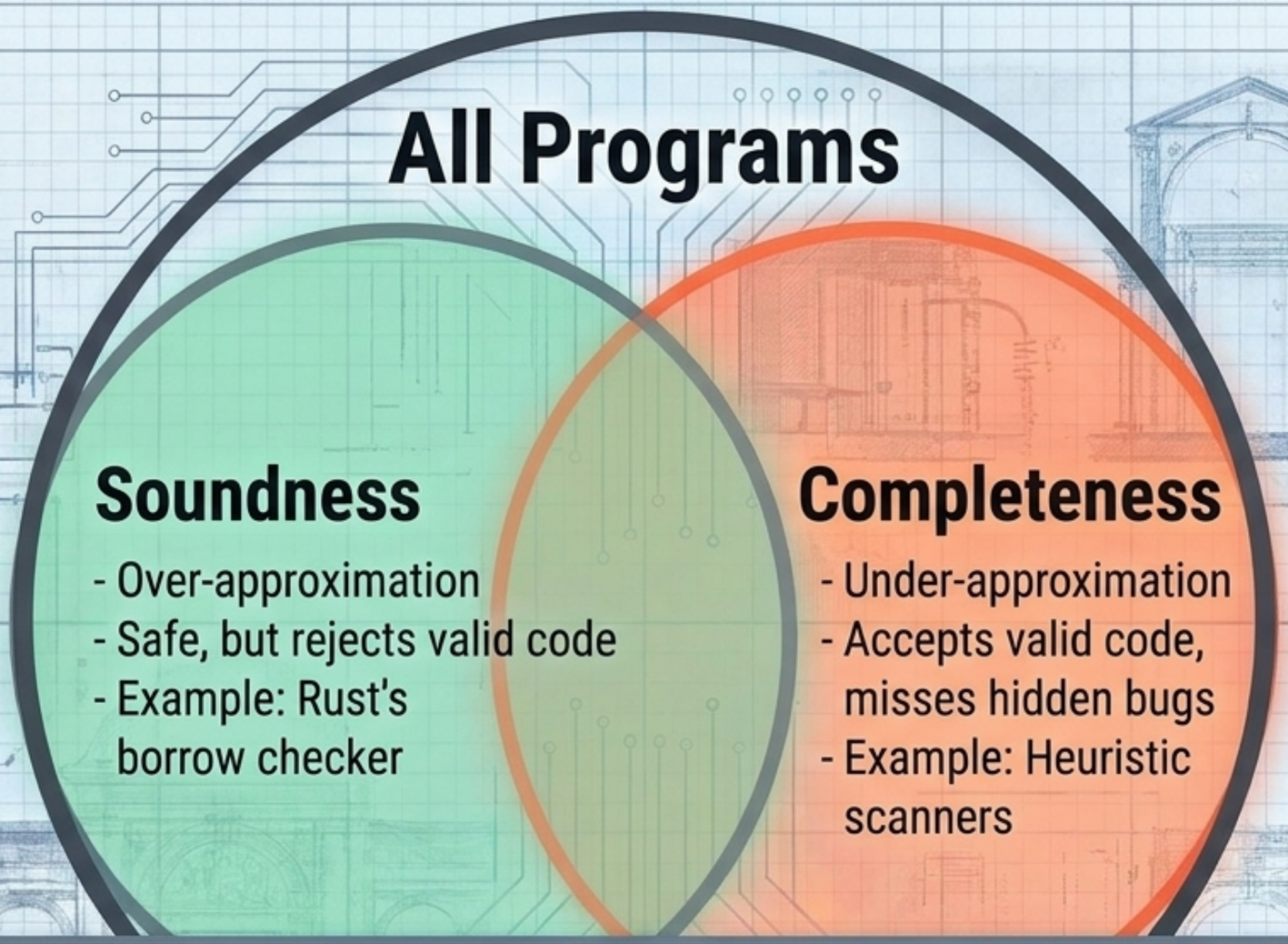
Gemini Notebook



The Loop-Snooper Contradiction: `paradox(P)` is designed to invert the output of the halting detector `halts(P, P)`.

The Result: A universal halting detector cannot exist. It destroys itself mathematically when forced to inspect its own negation.

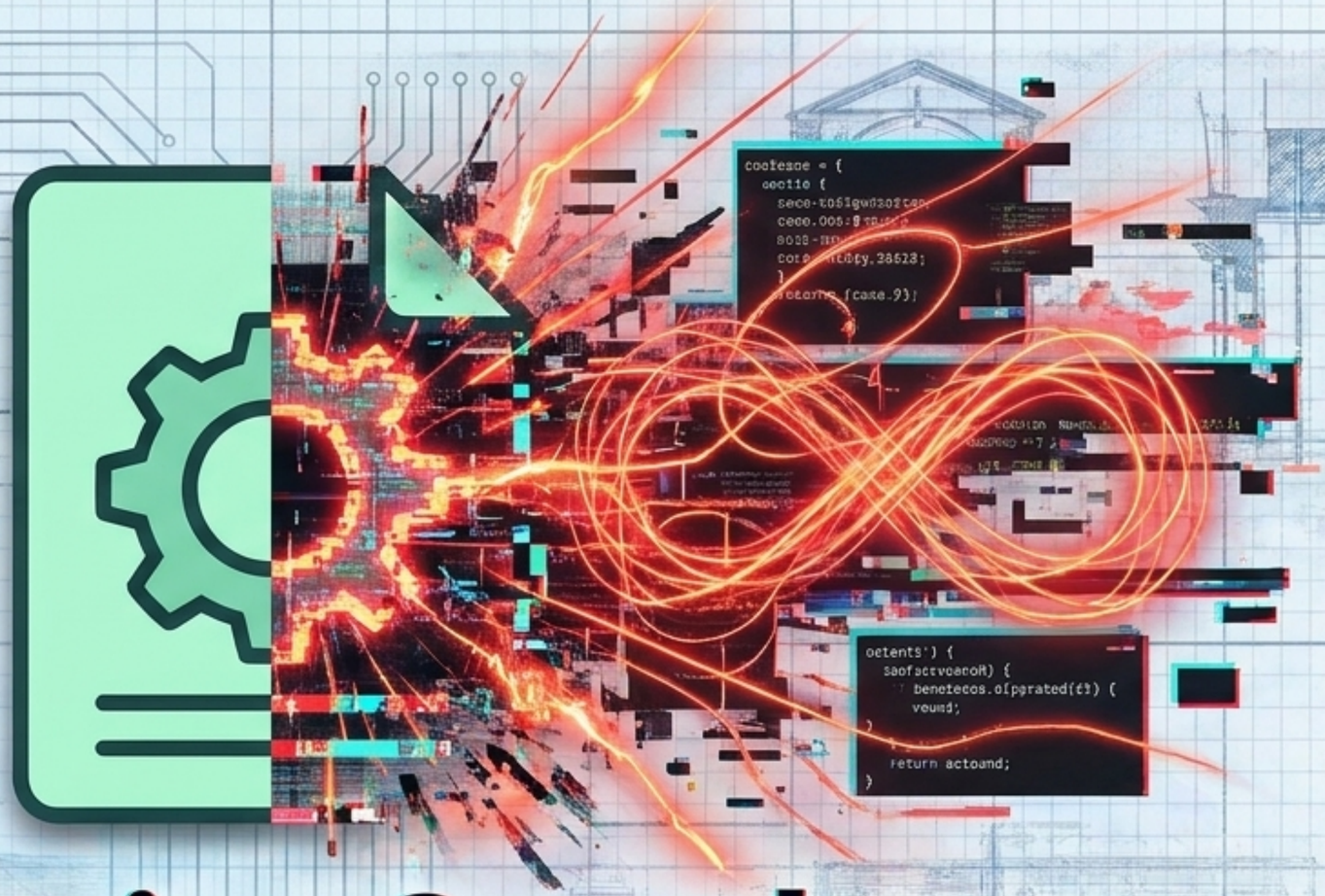
Gemini Notebook



Rice's Theorem (1953): Every non-trivial semantic property of a program is undecidable.
The Verification Trap: You must choose. You can be Sound OR Complete.
You cannot mathematically have both.

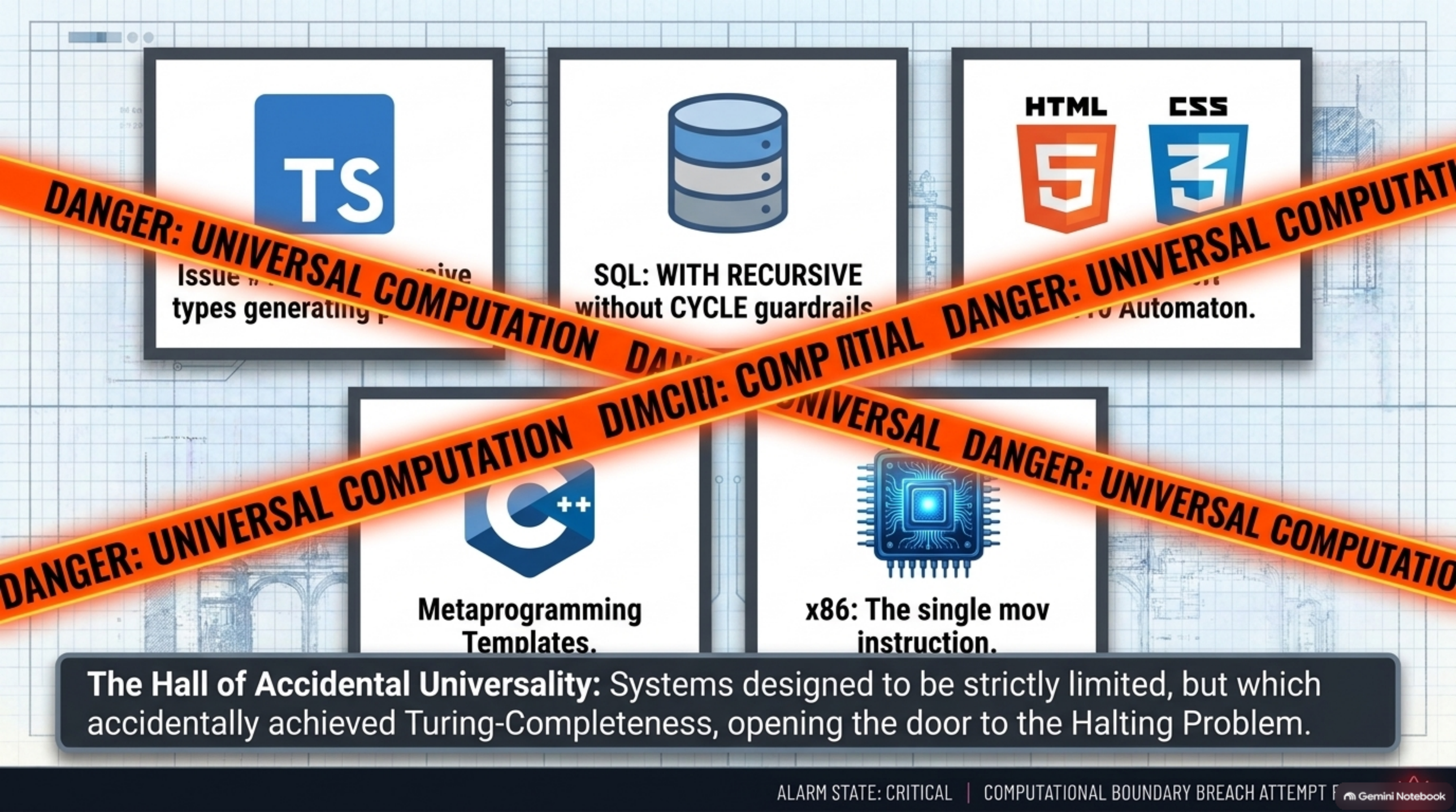
Gemini Notebook

Section 3



Accidental Turing Completeness: When Just Config Goes Infinite

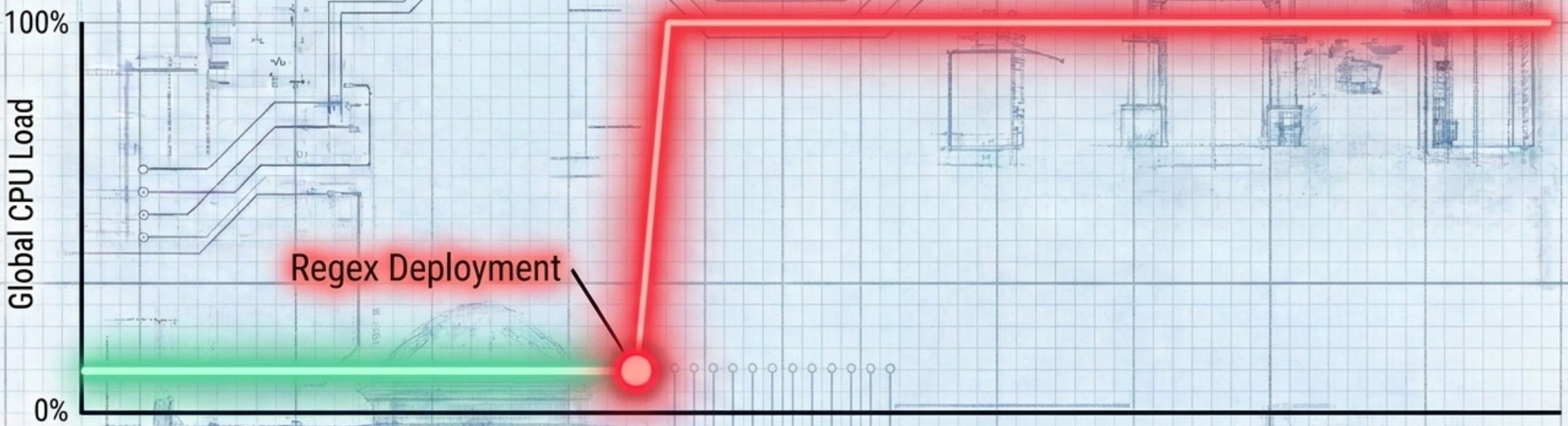
Gemini Notebook



The Hall of Accidental Universality: Systems designed to be strictly limited, but which accidentally achieved Turing-Completeness, opening the door to the Halting Problem.

Gemini Notebook

Section 3: The Regex Trigger Event



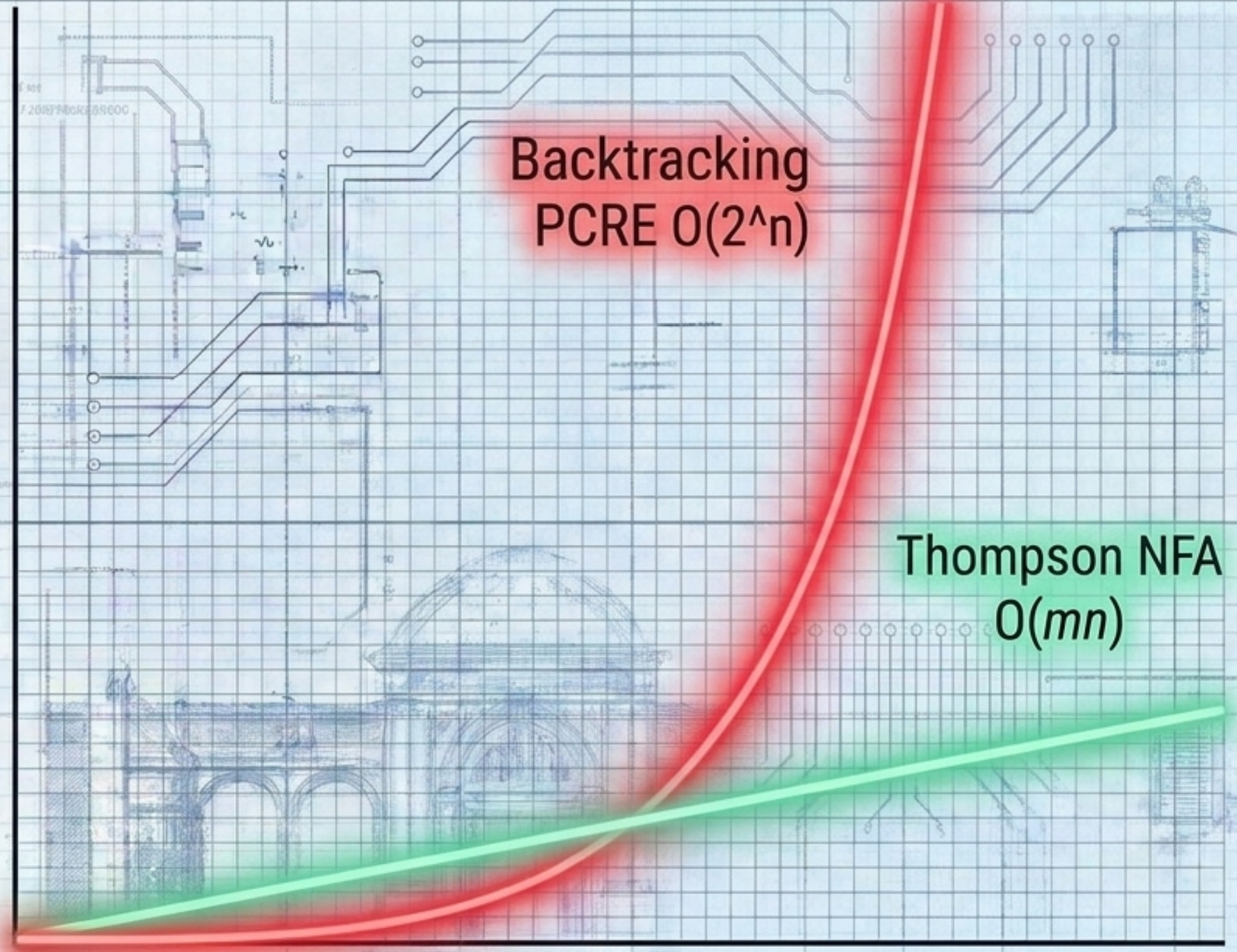
July 2, 2019 — The 27-Minute Global Outage

The Trigger: A single WAF regex deployed to target XSS: .*(?:.*=.*)

The Flaw: Catastrophic backtracking (CWE-1333) triggered by nested quantifiers.

The Result: Core proxying, CDN, and WAF functionality offline worldwide due to instantaneous, global CPU exhaustion.

Gemini Notebook



The Algorithmic Root Cause

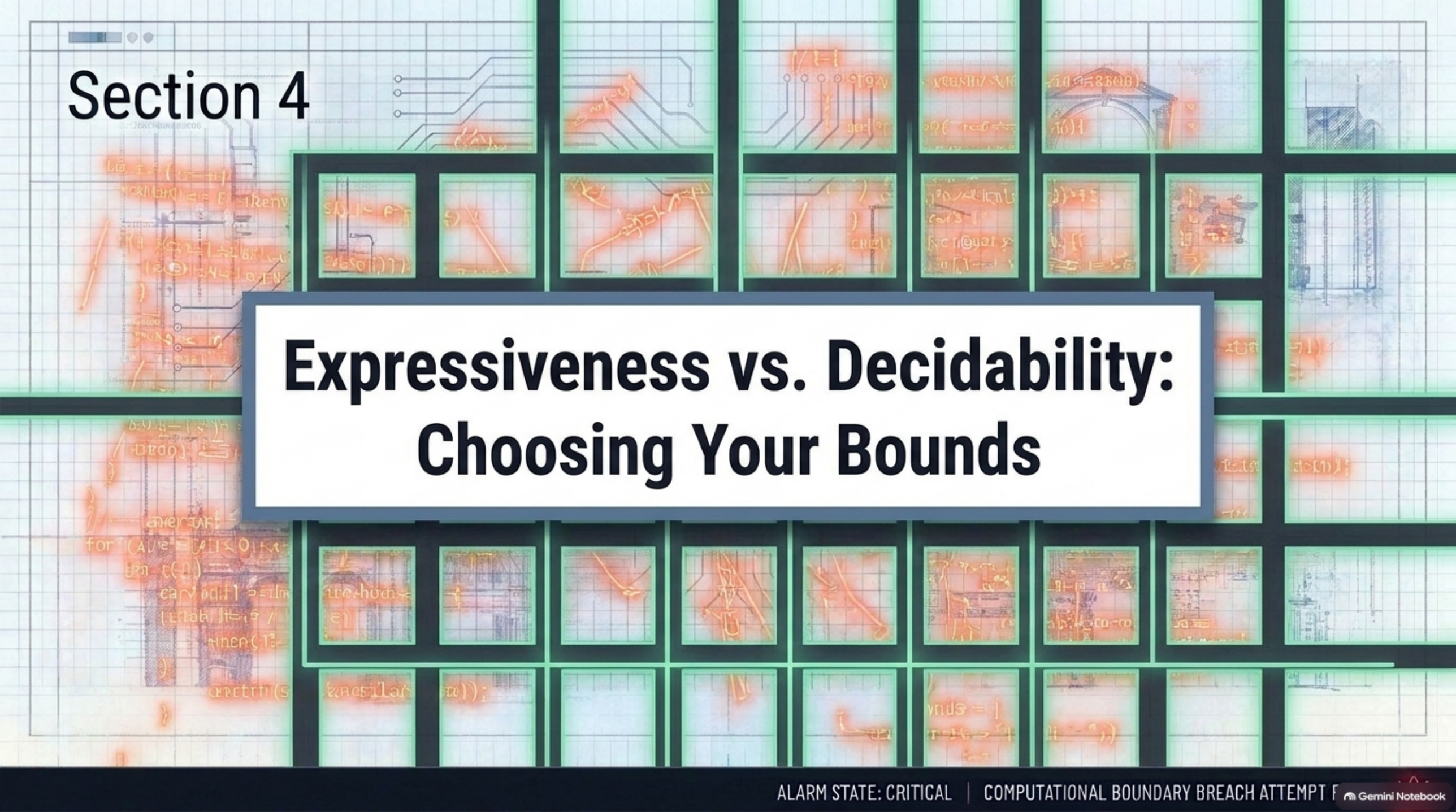
PCRE engines attempt to guess every possible path on crafted inputs, plunging into exponential time complexity.

A perfectly tiny input yields infinite processing destruction.

Gemini Notebook

Section 4

Expressiveness vs. Decidability: Choosing Your Bounds



ALARM STATE: CRITICAL | COMPUTATIONAL BOUNDARY BREACH ATTEMPT

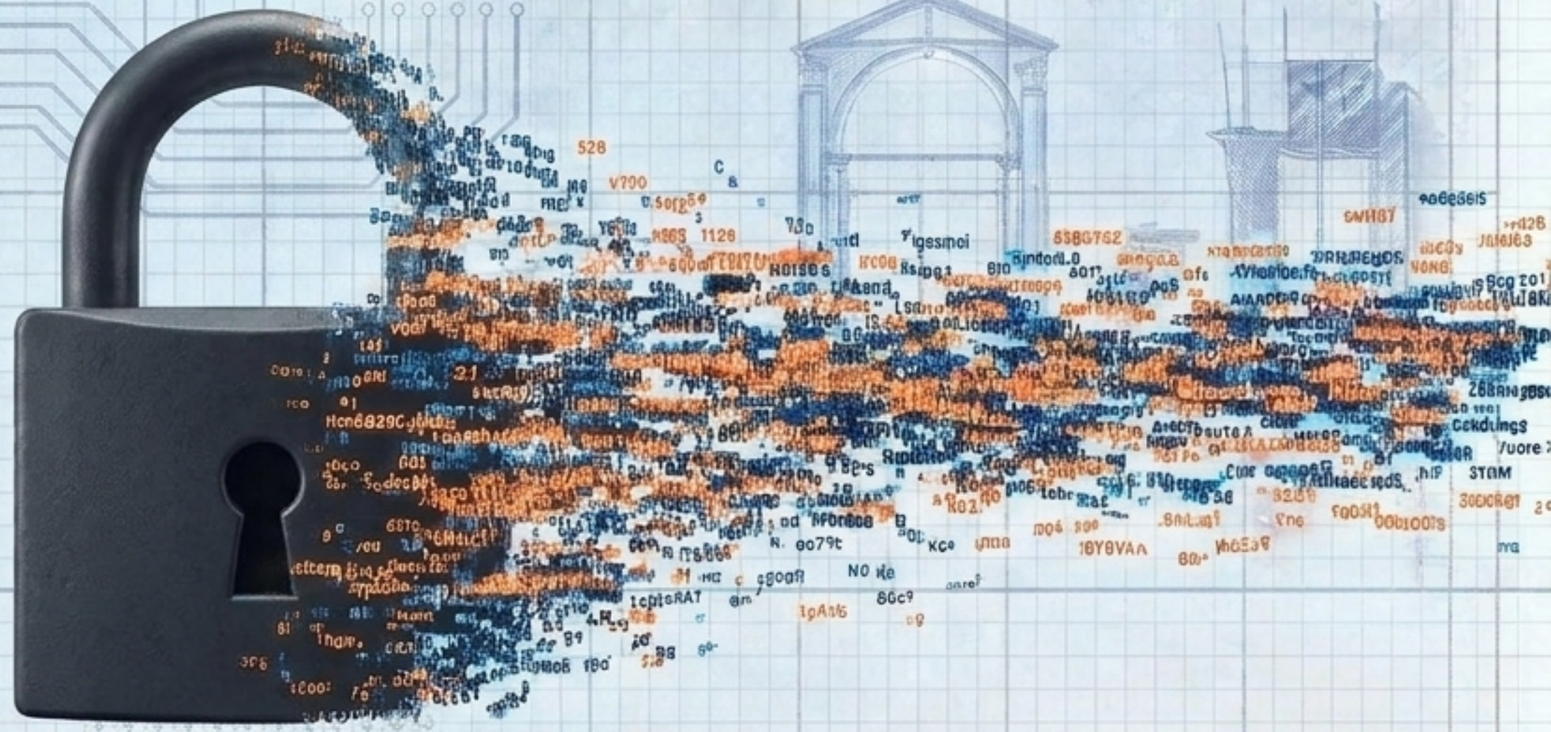
Gemini Notebook

Section 4: Expressiveness vs. Decidability Matrix

Technology	Time Bounds	Loop Semantics	Primary Use Case
Unrestricted (Turing-Complete)	eval(), PCRE Regex	Undecidable / Infinite	General purpose (High Risk)
Gas-Metered VMs	Ethereum EVM	Metered (halts when gas=0)	Smart Contracts
Bounded Model Checkers	Lamport's TLA+ / TLC	Finite-state exploration	Distributed Systems
Total Config/Policy	Google CEL, OPA Rego	Guaranteed termination	Fast, safe predicate logic
Linear-Time Automata	RE2, Rust regex	Strict O(n) guarantees	WAFs, untrusted text
Kernel Verifiers	Linux eBPF	Verified bytecode, strict bounds	Kernel observability

Gemini Notebook

Section 5

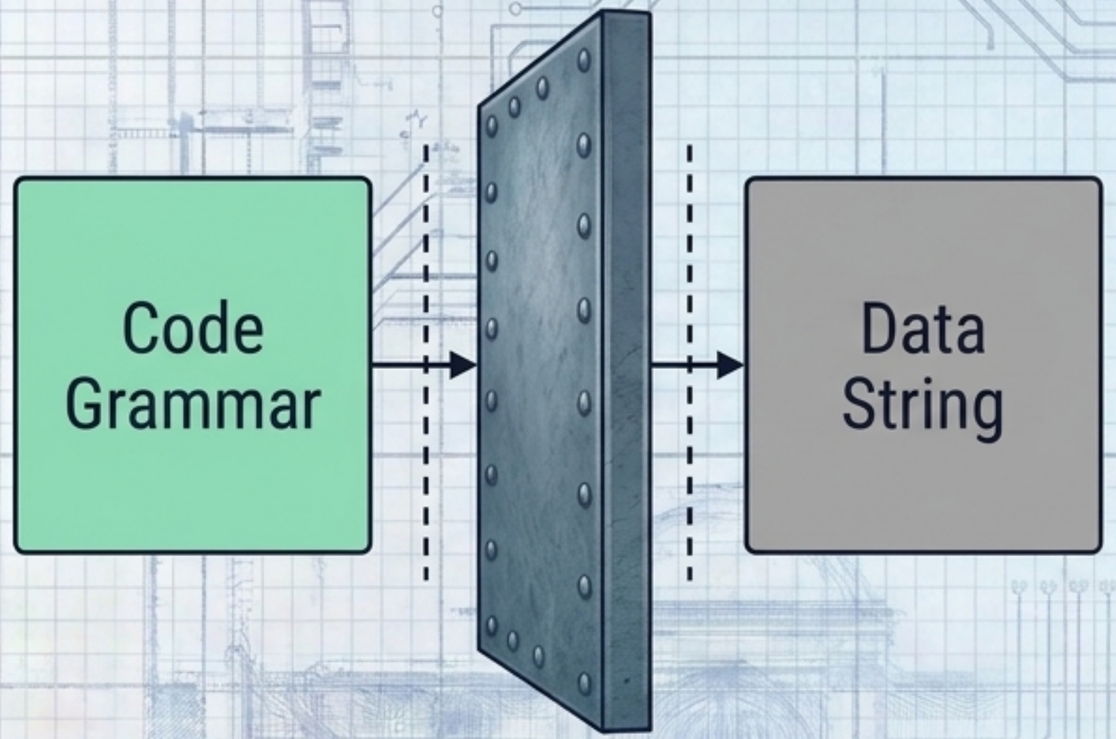


Prompt Injection & The Halting Problem

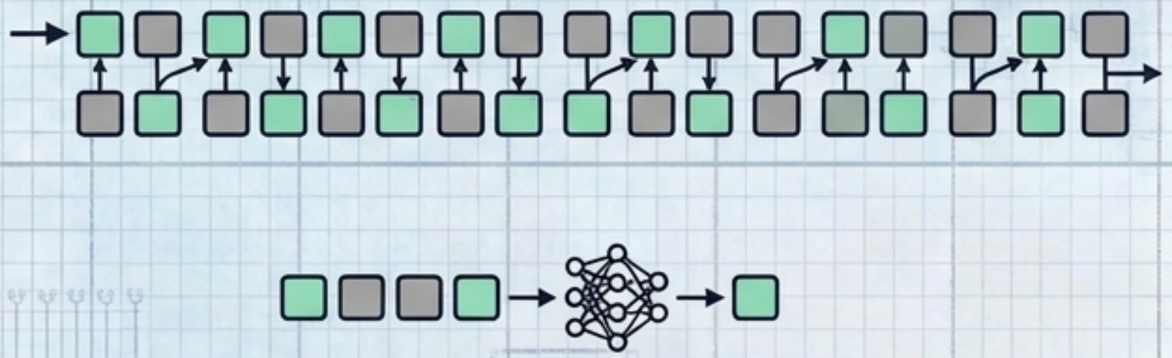
ALARM STATE: CRITICAL | COMPUTATIONAL BOUNDARY BREACH ATTEMPT

Gemini Notebook

SQLi - Decidable



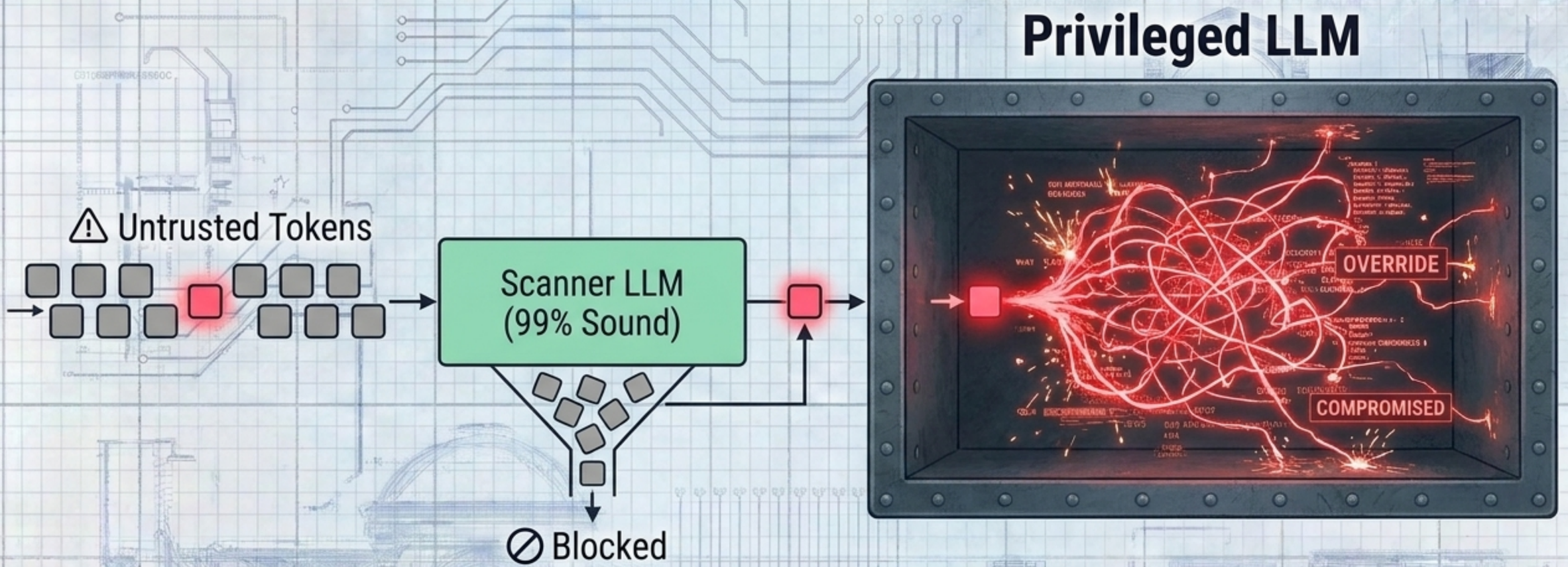
LLM - Inherently Confusable



The Grammar Divide: Why Prompt Injection isn't SQL Injection.

LLMs are an inherently confusable deputy. There is no structural boundary between an instruction and user data. There is only the prediction of the next token.

Gemini Notebook



The Guardrail LLM Fallacy: Why AI cannot secure AI.

Simon Willison: If we tried to solve SQL injection with a solution that only worked 99% of the time, none of our data would be safe.

By Rice's Theorem, a 100% sound semantic filter is mathematically impossible. Security demands deterministic capability sandboxes, not heuristic AI judges.

Gemini Notebook

Section 6





The Vibe-Coder Trap: Omniscient-AI Fallacies

Gemini Notebook



The Zero Bug Prompt: Asking an LLM to guarantee zero injections (Impossible via Rice's Theorem).



ReDoS Roulette: Deploying PCRE regexes with nested quantifiers (CWE-1333).



The Eval() Filter: Trusting dynamic code execution to sanitize user inputs.



Unbounded Retries: Writing while/useEffect loops without termination limits (CWE-835).



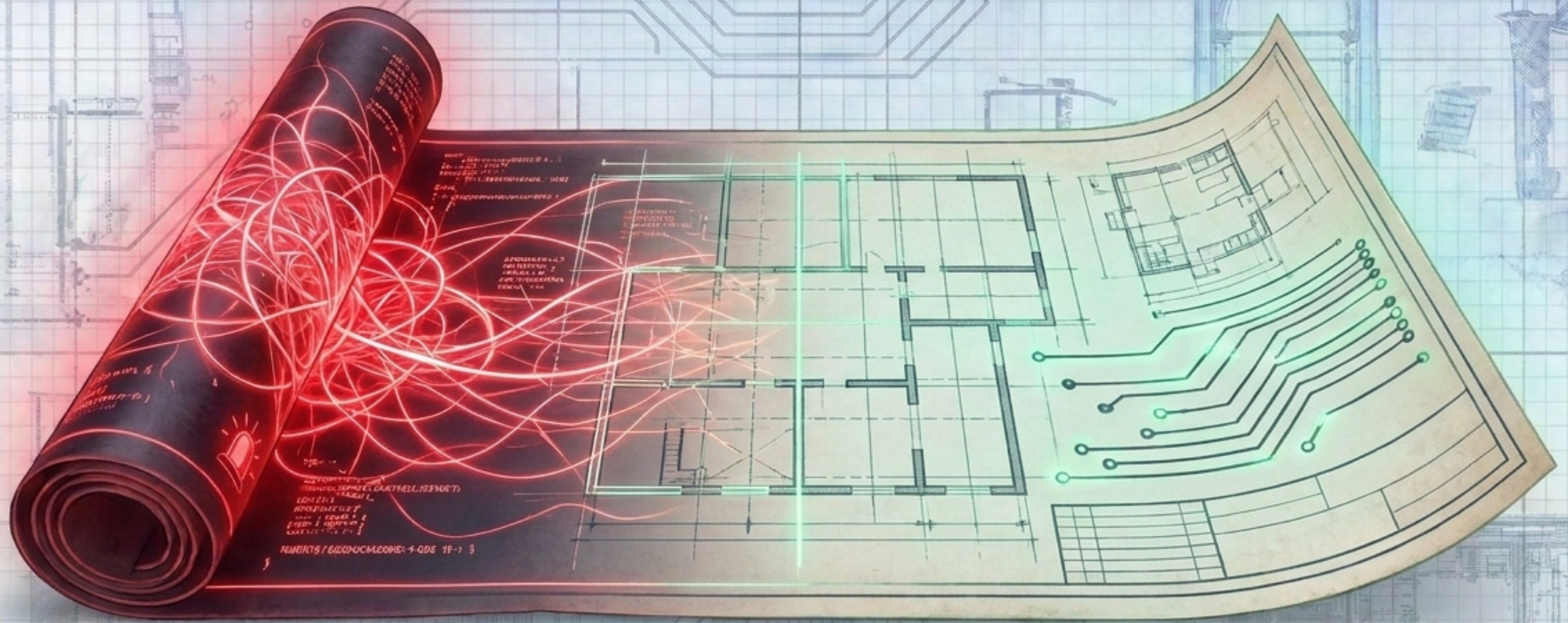
Runaway Agents: Tool-use loops consuming tokens without circuit breakers (OWASP LLM10).



The Compiler Hang: Generating recursive TypeScript types that crash the build.

Gemini Notebook

Section 7



The Playbook: Engineering for Decidability

Gemini Notebook

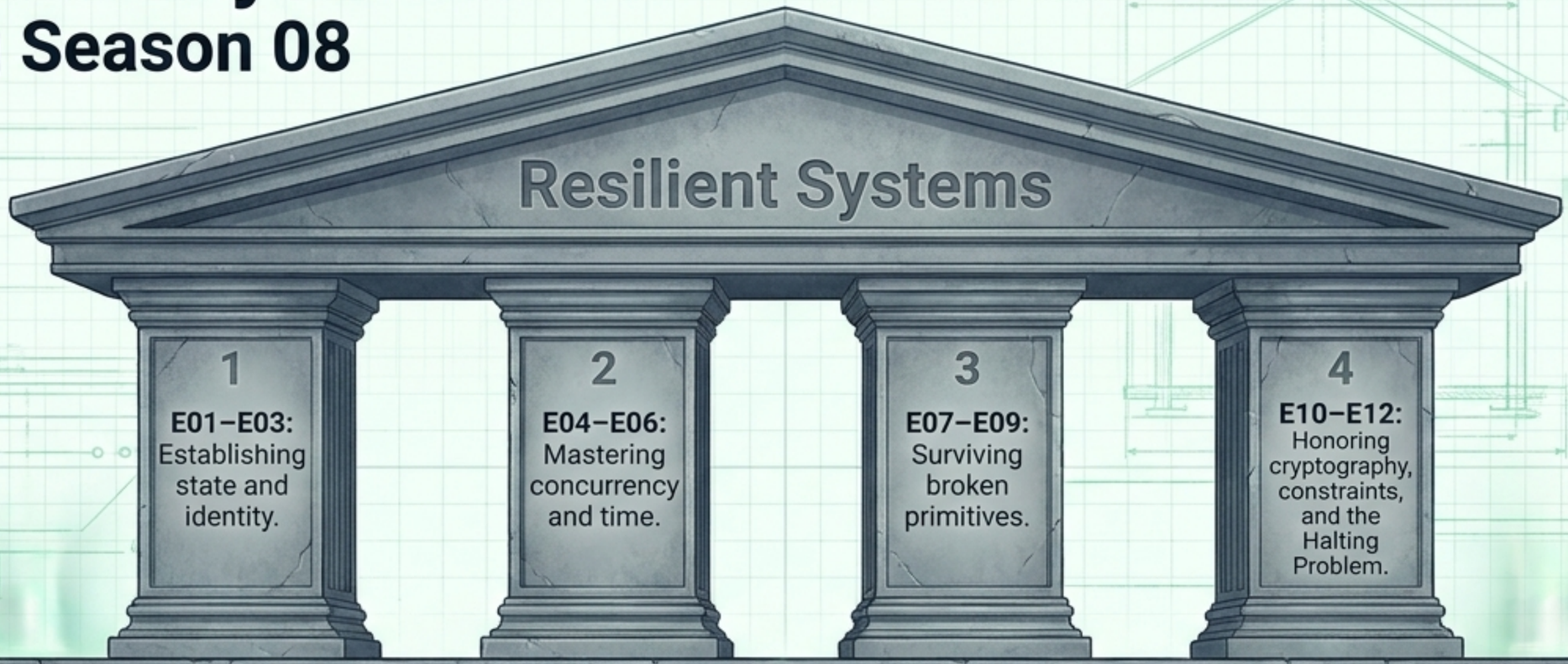
The Termination & Decidability Ruleset

CLAUDE.md / AGENTS.md

- **Prefer Decidability:** Mandate RE2 over PCRE; Google CEL/OPA Rego over eval().
- **Enforce Limits:** Any SQL WITH RECURSIVE must carry a strict CYCLE + LIMIT.
- **Agent Circuit Breakers:** Mandate hard maxIterations, AbortSignal timeouts, and strict token/cost budgets.
- **YAGNI Guardrails:** Avoid over-engineering. Do not reach for TLA+ or custom DSLs when simple bounded configurations will suffice.

Gemini Notebook

Capstone Synthesis
Uniting Season 08



Capstone Synthesis: Uniting Season 08

Thesis: Great engineering is not about infinite power; it is about designing brilliant constraints.

⚠ ALARM STATE: CRITICAL | COMPUTATIONAL BOUNDARY BREACH ATTEMPT FAILED

Gemini Notebook