

# L'API e l'Architetto

Anatomia del Vibe Coding  
tra l'Idea Pura e la Macchina

Un'analisi del caso studio  
CodiceConvivialePlus.

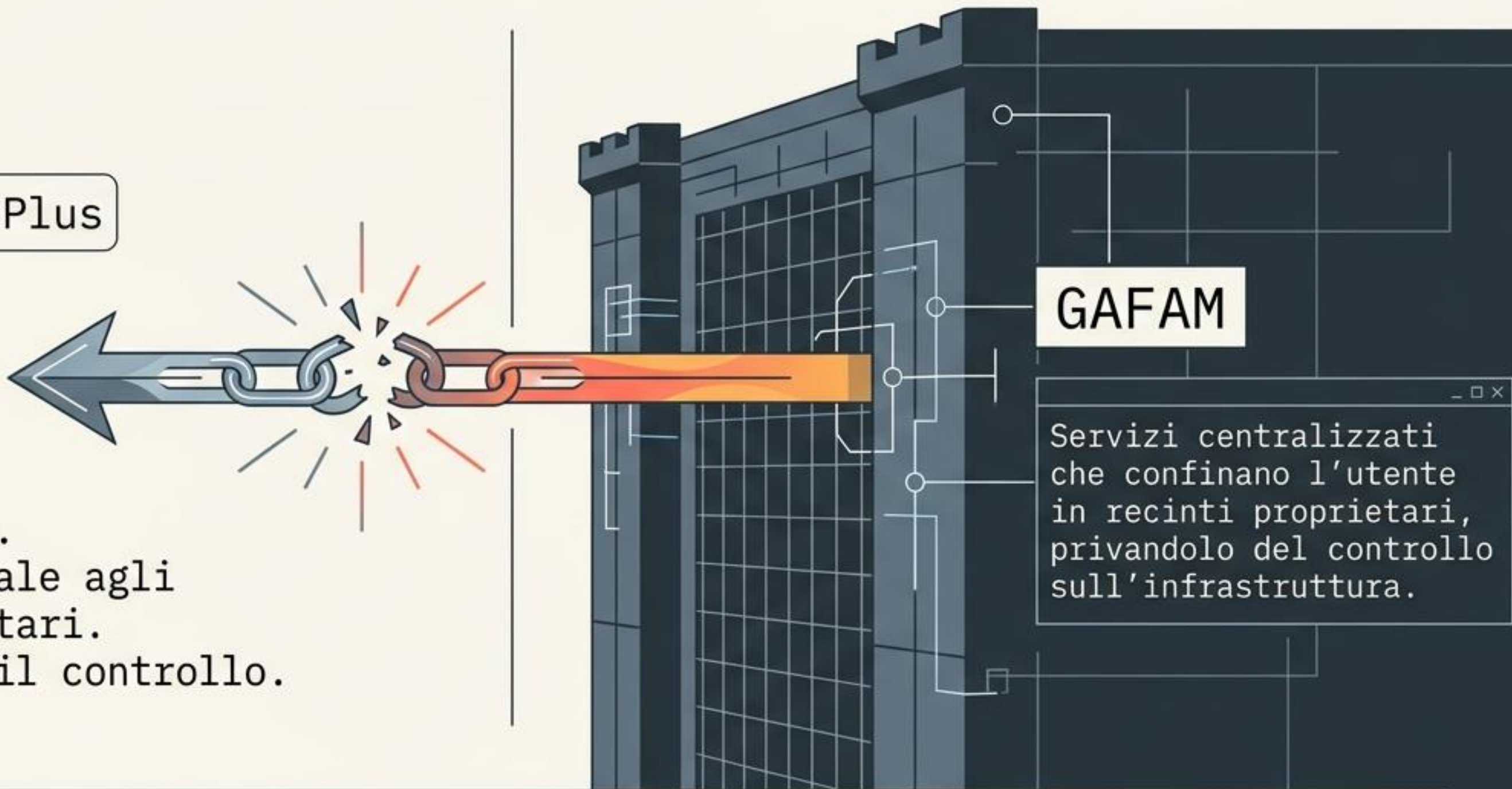
Da un'idea di  
Marco Guastavigna

# Fuga dal Recinto Oligopolistico

CodiceConvivialePlus



Sovranità Digitale.  
Un'alternativa locale agli  
assistenti proprietari.  
L'utente riprende il controllo.



L'obiettivo è permettere a chiunque di implementare e verificare un'idea con un'Intelligenza Artificiale Generativa gestita interamente in locale.

# Il Paradigma dello Split View.

**Vibe Coding:** Lo sviluppo software in cui l'umano esprime il ritmo e l'idea concettuale a parole (Linguaggio Naturale), delegando la stesura materiale della sintassi all'IA.

Workspace/Chat



Idea concettuale...

Lo spazio cognitivo cambia: il focus si sposta dall'editor di testo vuoto alla chat conversazionale. La barriera d'ingresso per testare un prototipo crolla.

```
<div class='container'>
```

```
function init() {  
  ...  
  ...  
}
```

Sandbox

style.css

```
function init() {  
  ...  
  ...  
}
```

style.css

# Il Dibattito: Ideatore vs. Meccanico

## VibeCodingFan

“ Il codice è un dettaglio esecutivo. Il valore risiede nell'ideazione.

Ruolo dell'Utente

Regista cinematografico (guida la visione).

Percezione della Frizione

Dettagli operativi minimi, facilmente superabili.

## AntiVibeCoding

“ L'astrazione totale è un'illusione. Le competenze tecniche restano imprescindibili.

Ruolo dell'Utente

Meccanico/Sistemista (deve sapere come aggiustare la macchina).

Percezione della Frizione

Barriere strutturali che possono uccidere l'idea prima che nasca.

# L'Evocazione della Complessità

Crea una lavagna interattiva per disegnare con stili al neon

Cyberpunk Calc -> Evoca: Web Audio API.  
La matematica complessa dei sintetizzatori  
anni 80 gestita senza conoscere la sintassi.

Zen Pomodoro -> Evoca: Generazione Procedurale.  
Rumore bianco creato da algoritmi matematici  
in tempo reale.

RPG Task Quest -> Evoca: Logica di Gamification.  
Stati complessi e punti esperienza tradotti  
da una semplice richiesta concettuale.

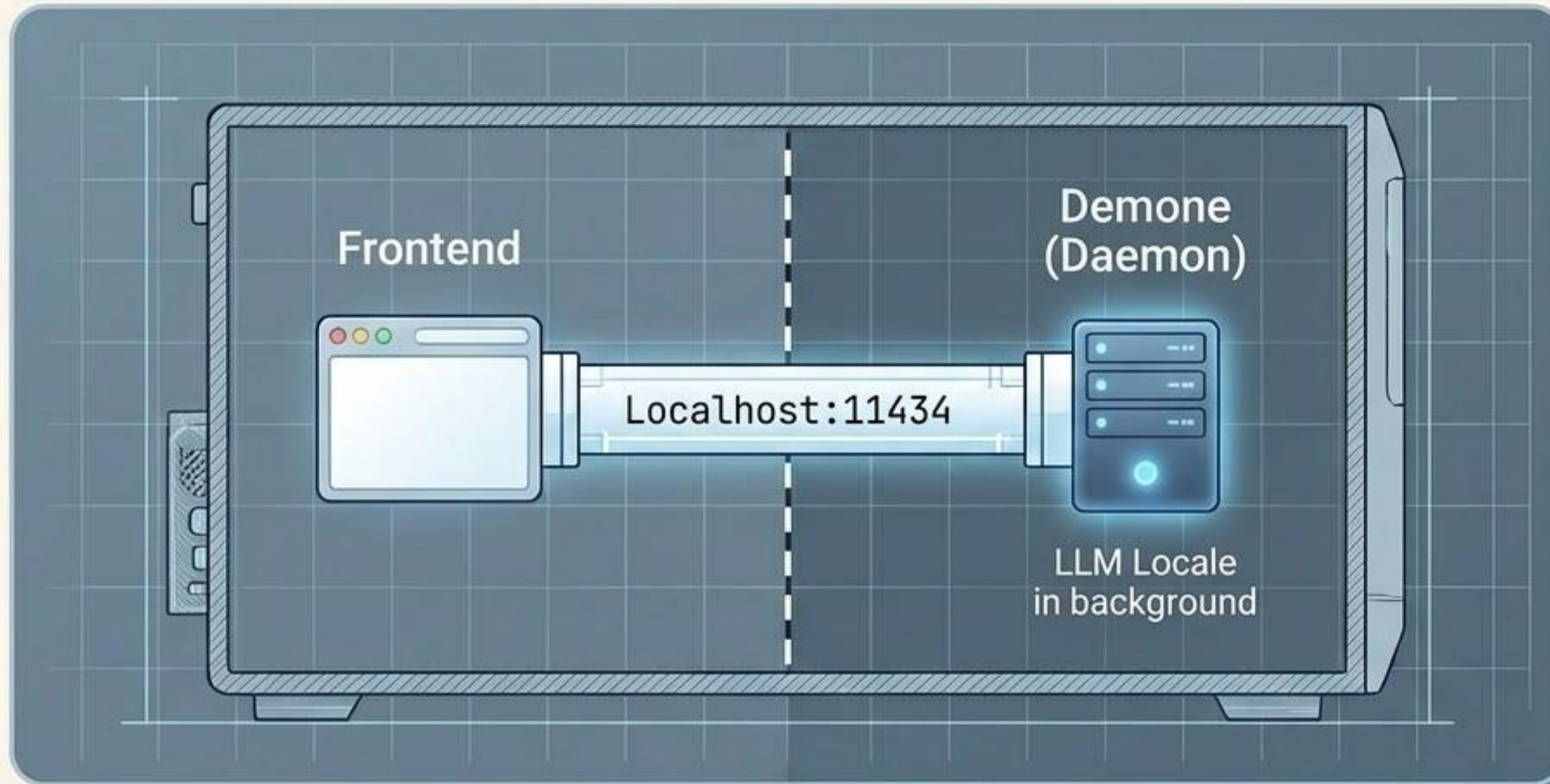
Il limite non è più: So come scriverlo?,  
ma: Riesco a immaginarlo?.

# L'Imbuto della Frizione Infrastrutturale.



**Il software non vive nel vuoto delle idee platoniche, ma in ambienti ostili e strutturati.**

# Ostacolo 1: La Topologia Invisibile.



L'utente deve sapere che Ollama non è solo acceso, ma risiede a un indirizzo e a una porta specifici all'interno della stessa macchina.

Questo non è linguaggio naturale. Richiede alfabetizzazione sistemistica prima ancora di poter formulare un prompt.

# Ostacolo 2: Sbloccare il CORS

CORS (Cross-Origin Resource Sharing).  
Il portiere severo del browser che blocca  
le richieste incrociate di default.

Browser Security Wall



Interfaccia Grafica (GUI)  
tramite LM Studio.

L'abilitazione CORS è gestita  
automaticamente da bottoni e menu.

```
> OLLAMA_ORIGINS=*
```

Terminale / Riga di Comando  
tramite Ollama.

L'utente deve digitare  
variabili d'ambiente di sistema.



**Critica:** L'ideatore si trova costretto a manipolare variabili d'ambiente.  
Se fallisce, deve saper leggere un log di rete (F12).

The Vibe

# Ostacolo 3: La Fragilità del Markdown.

The Code

I modelli LLM sono macchine probabilistiche. Se manca l'accento grave isolato (Backtick), il sistema non sa dove finisce la chiacchierata e inizia il codice.

Ecco il codice per la tua applicazione:  
```html



La regitiva congelazione na come antrivamenio autoritativa critique. L'ideatore deve indossare la **tuta da meccanico** e forzare il **rendering a mano** nel tab HTML, distinguendo il testo dall'inizio del Doctype.

Ecco il codice per la tua applicazione:  
html

Zero Assoluto

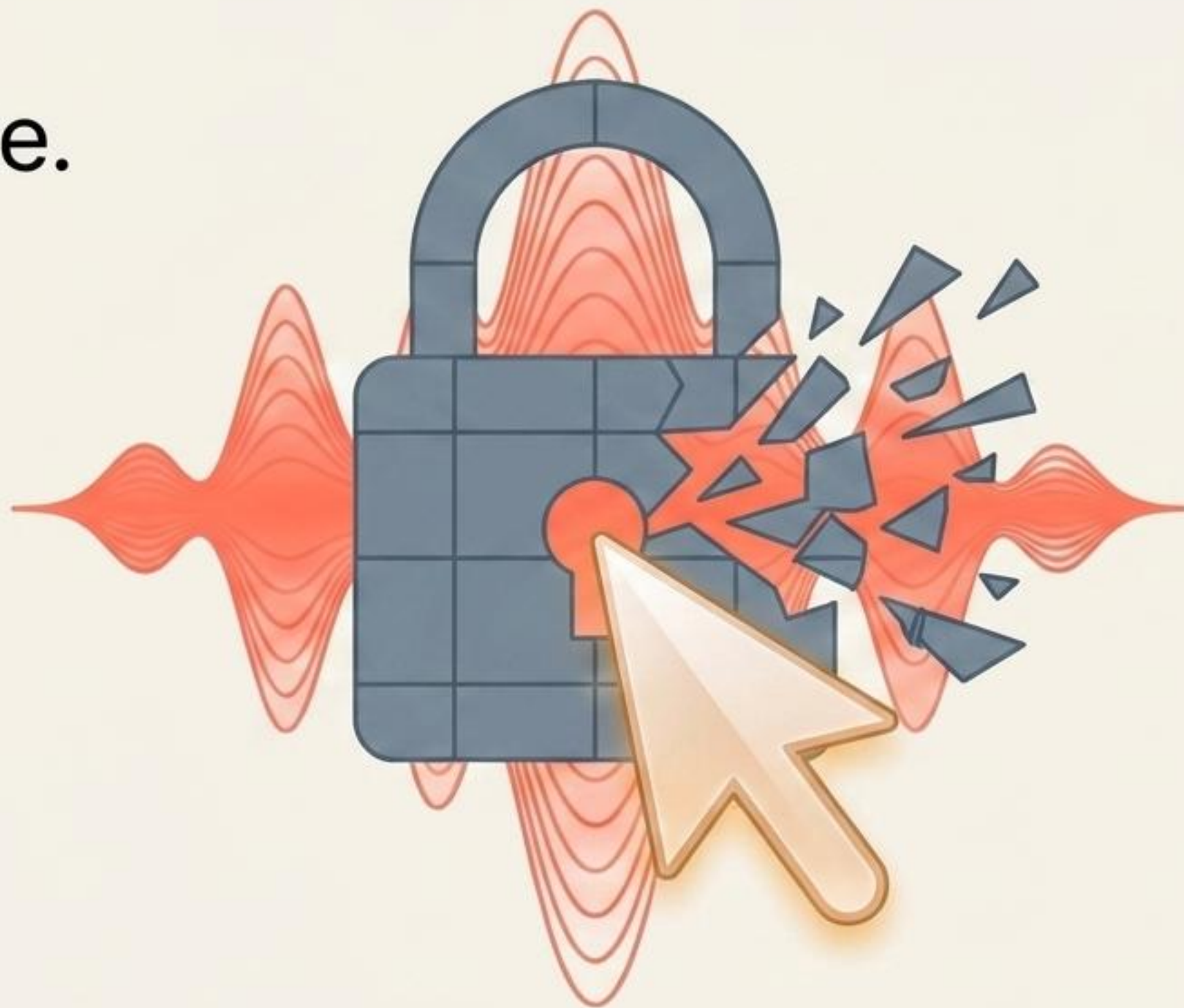
WARNING

# Ostacolo 4: Le Regole dell'Ambiente.

## Audio Context:

Il regista audio del browser.

Il browser blocca l'audio di default per evitare suoni aggressivi all'apertura della pagina.



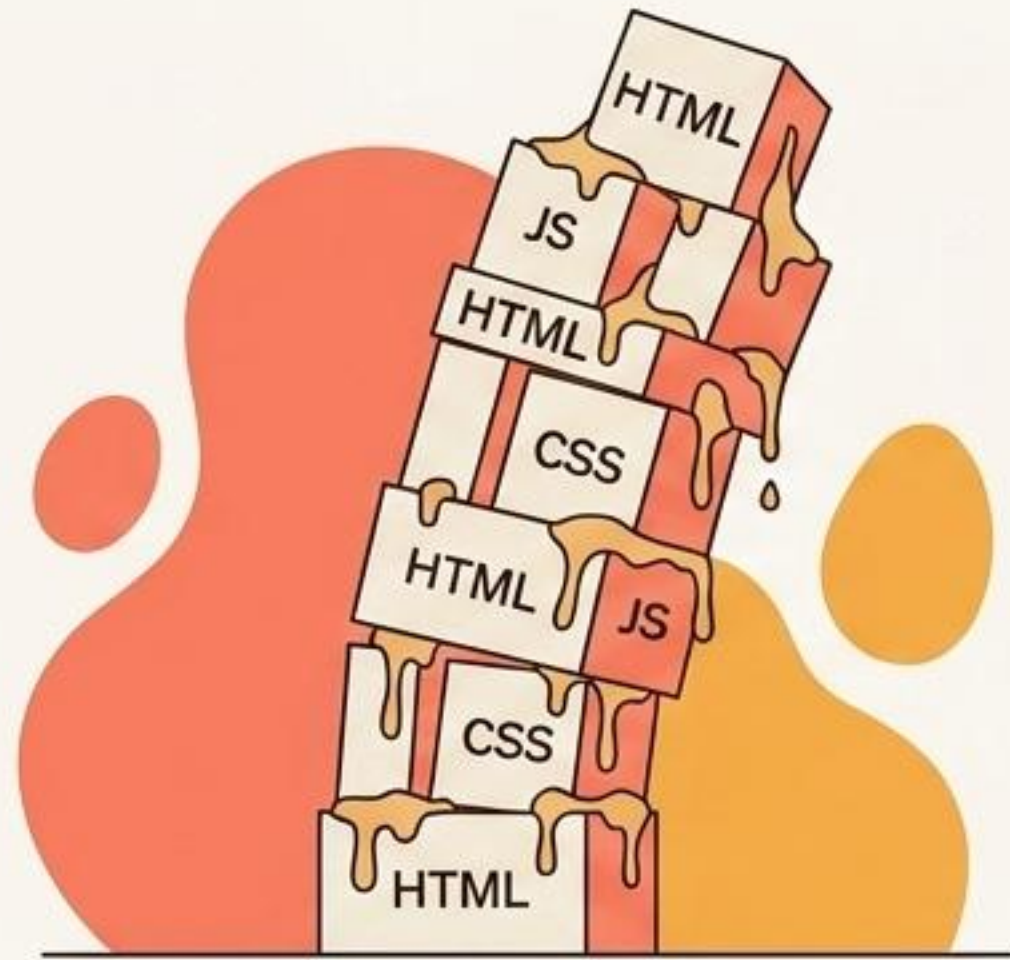
L'illusione della magia autonoma si scontra con la necessità di un'azione umana esplicita (un clic fisico) per sbloccare la funzione.

Il dispositivo impone le SUE regole.

L'intenzione mentita l'unione autonoma con la necessità di un'azione umana esplicita (un clic fisico).

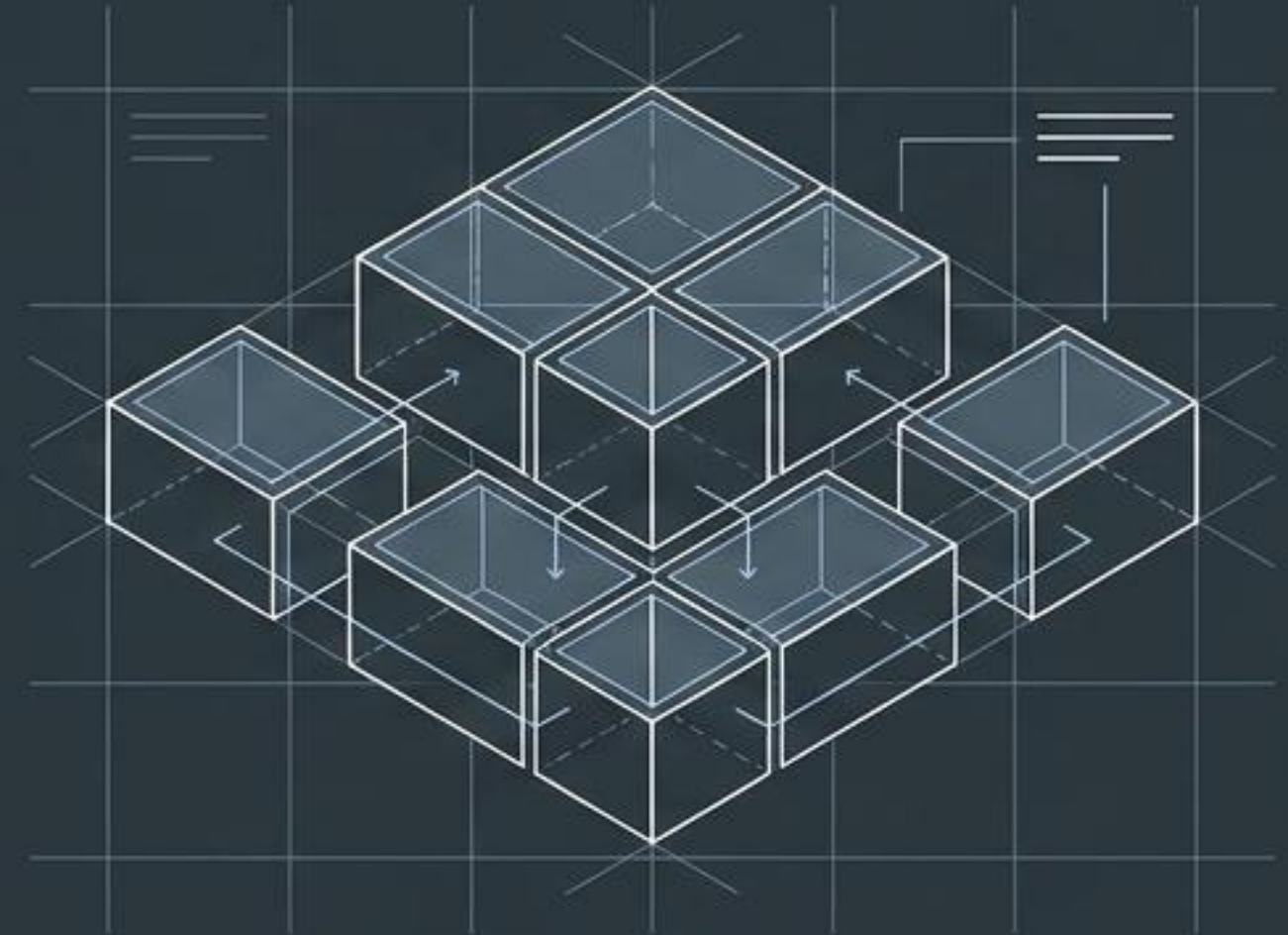
Il dispositivo impone le SUE regole.

# L'Ambiente Monolitico: Portabilità vs. Debito Tecnico.



**Il Palazzo con la colla a caldo**

**Pro:** Indipendenza totale. Nessuna dipendenza, nessun pacchetto da installare. Il veicolo perfetto per la distribuzione via file singolo.



**Architettura Modulare**

**Contro:** Un incubo di manutenzione. Lo stato globale si mischia agli stili. Regge finché è alto 20 cm, poi crolla.

# Amnesia da Finestra di Contesto

La quantità massima di dati che l'IA può tenere a mente.



Man mano che il file monolitico cresce di migliaia di righe, l'IA perde coerenza, dimentica i pezzi precedenti e genera errori a cascata nello spazio dei nomi globale.

# L'Architettura del Vibe Coding.

Livello 1: L'Esperienza



Livello 2: Il Confine



Livello 3: Le Fondamenta



L'idea pura non esiste senza le sue fondamenta. Il Vibe risiede nel livello superiore, ma l'esecuzione dipende dall'integrità dei livelli inferiori.

# Sintesi: Il Direttore e la Cinepresa.

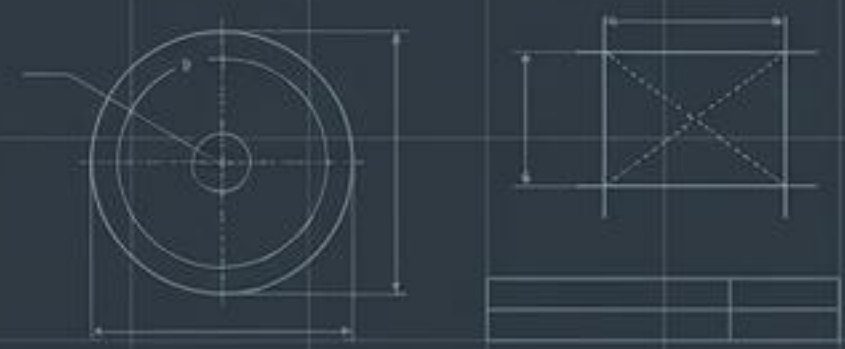
## The Vibe

Nascondere la complessità abbassa la barriera d'ingresso (la frizione iniziale è quasi zero), ma rende la complessità molto più difficile da risolvere quando il sistema si rompe.



## The Code

Mechanical architect  
strucort design  
terniogy handing



Il Vibe Coding non elimina l'ingegneria, ne eleva l'astrazione.  
L'utente diventa un regista, ma un regista che DEVE sapere  
come aggiustare la cinepresa se si inceppa la pellicola.

# Oltre l'illusione della Sabbia.



“Finché ci saranno limiti monolitici, CORS da sbloccare e markdown che si rompono, l'architetto avrà sempre bisogno di comprendere il codice per trasformare un prototipo di sabbia in un edificio solido.”

La transizione tra l'epoca dell'Ingegneria dei sistemi e il sogno dell'idea pura è appena iniziata.