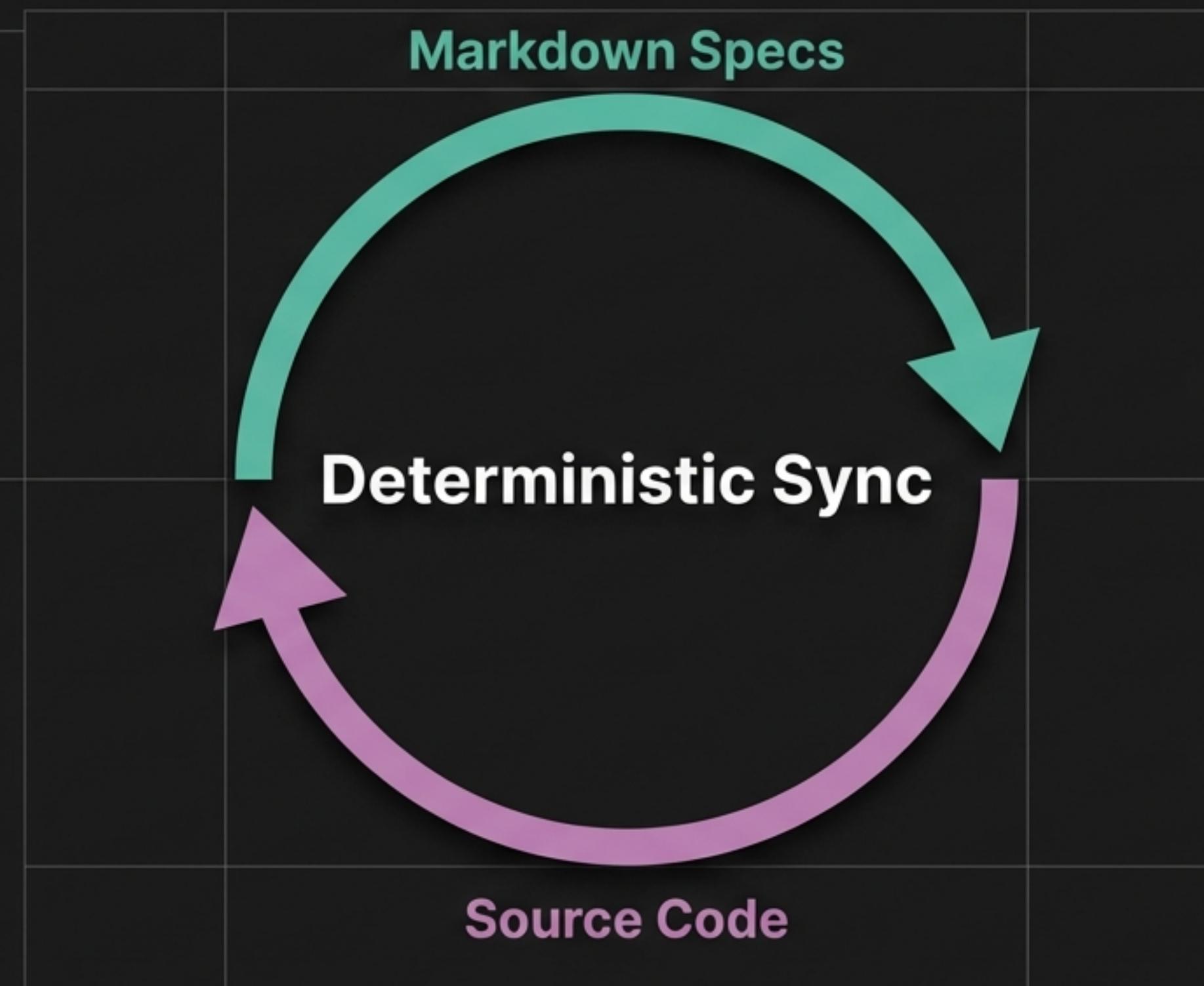


Spec-Driven Development in the AI Era

A systematic framework for shipping features in record time using Open Spec & Cursor.

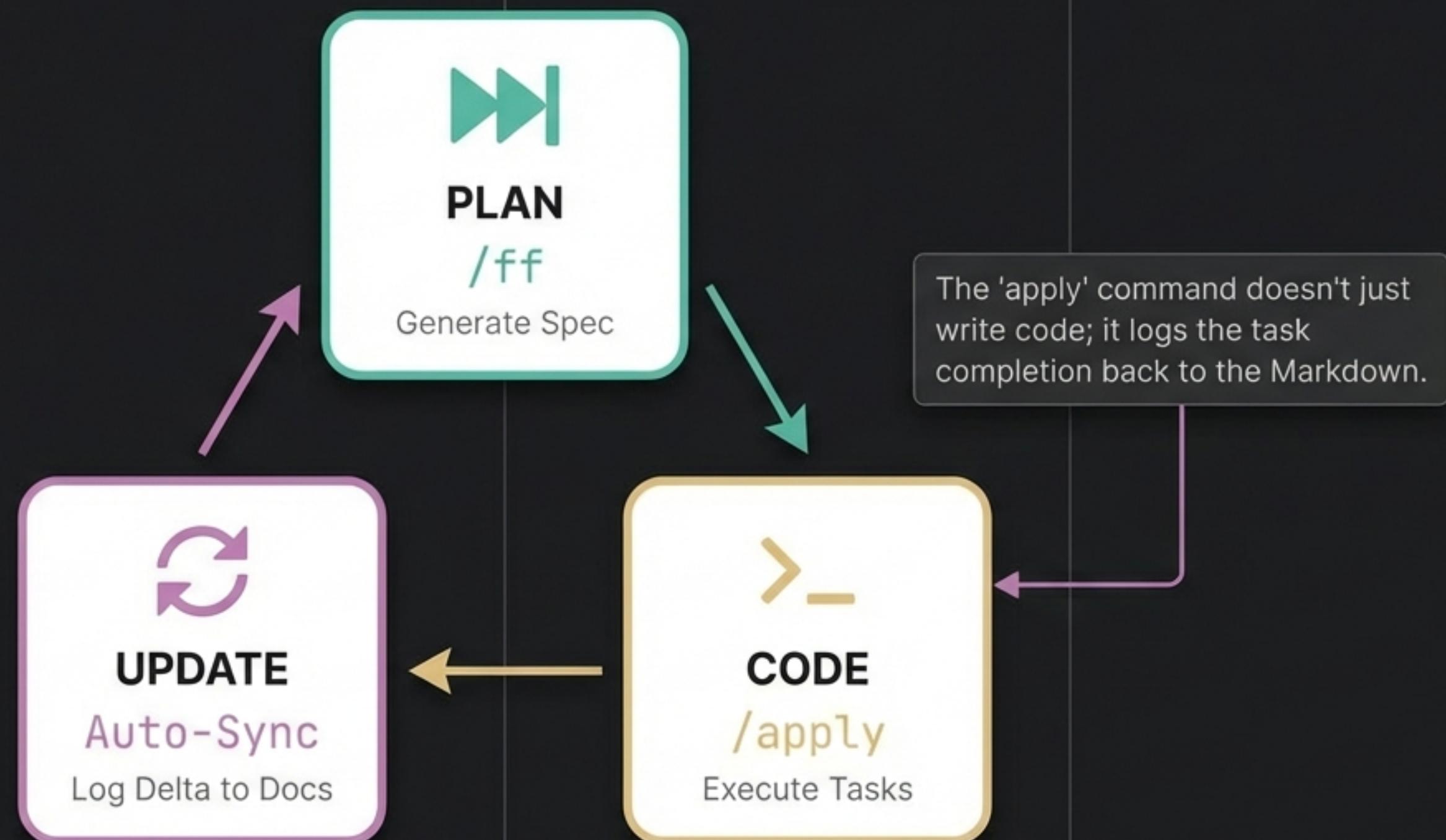


The Drift: Why 'Chatting with Code' Fails



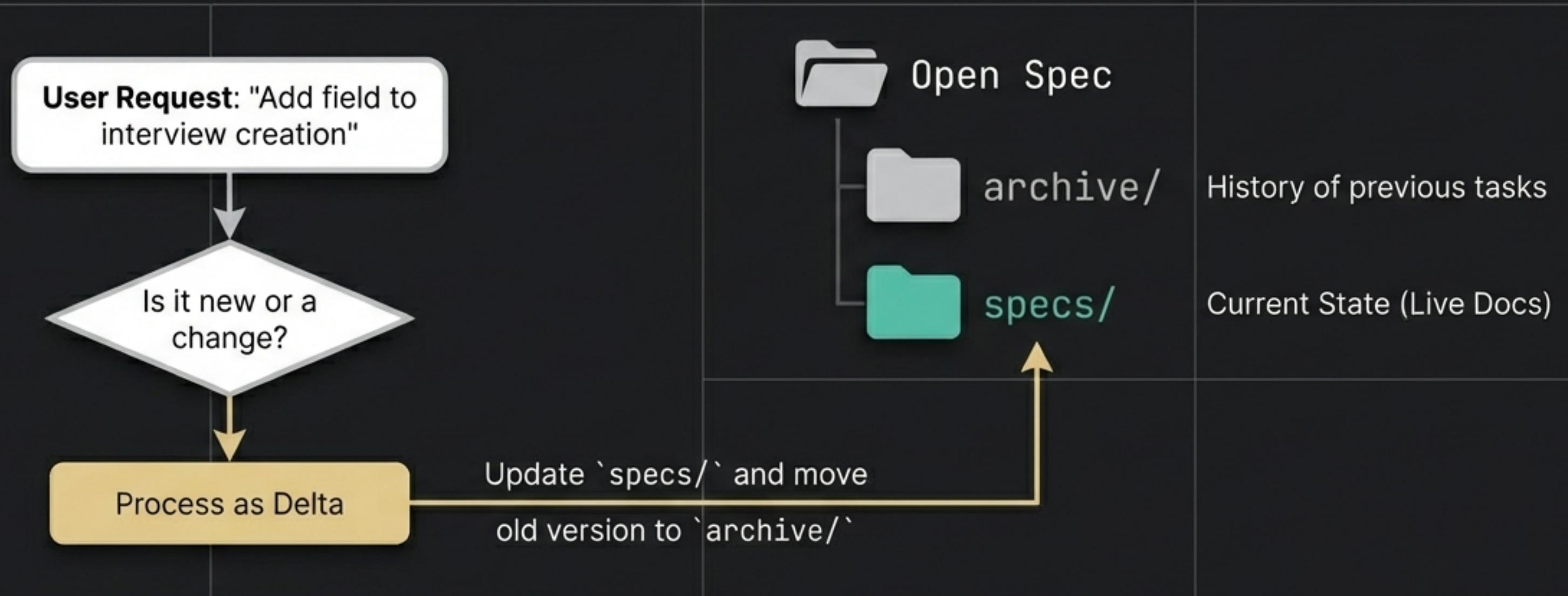
Problem: Without a framework, AI tools 🧠 act autonomously. They generate code improvements that your documentation never sees, rendering the "Plan" ⚙️ obsolete instantly.

The Solution: The Open Spec Loop



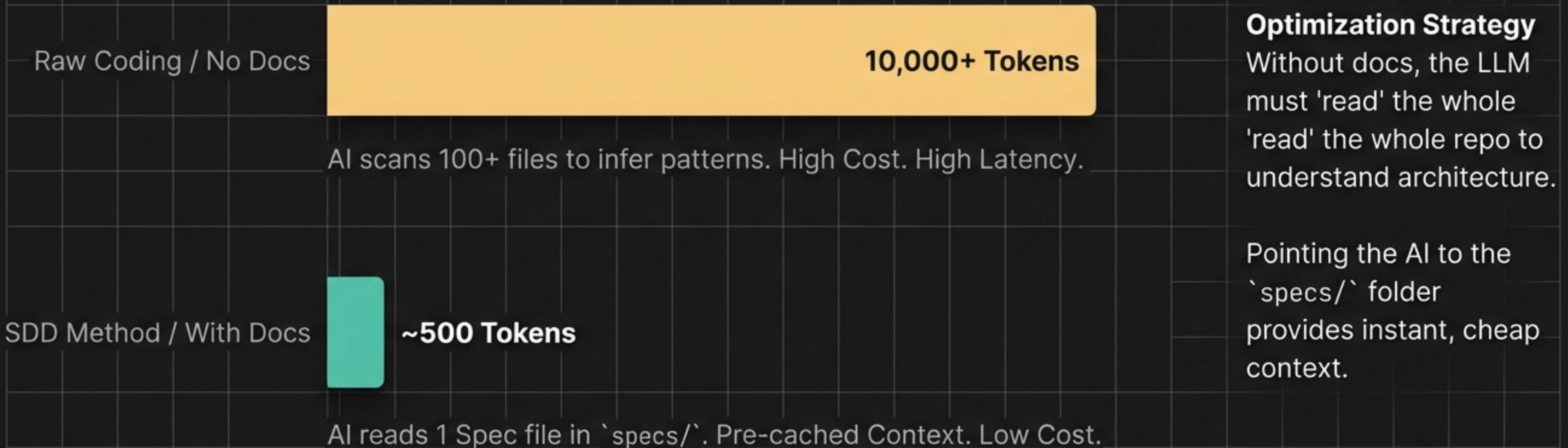
Managing the Delta

How to maintain the Single Source of Truth.



Automated Traceability: No manual Markdown updates required. The framework tracks whether a task was a one-shot or an iteration.

The Token Economy

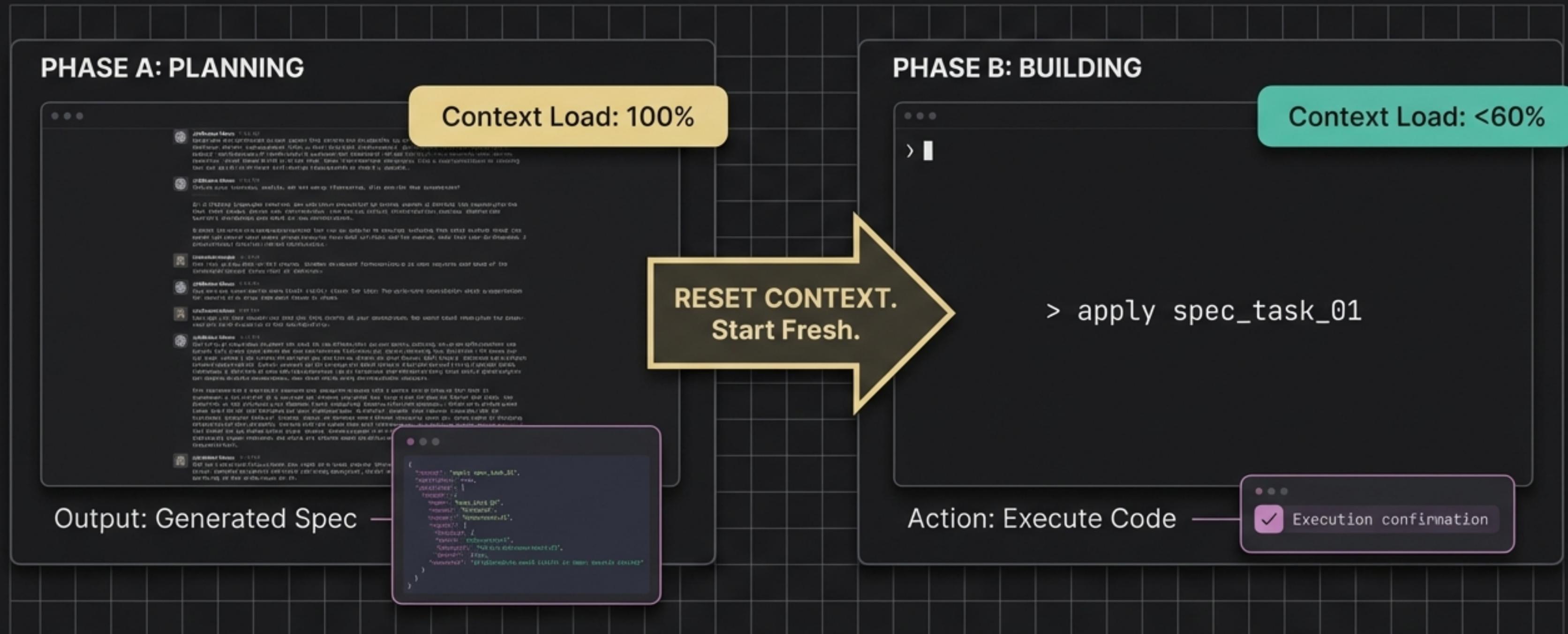


Optimization Strategy

Without docs, the LLM must 'read' the whole 'read' the whole repo to understand architecture.

Pointing the AI to the `specs/` folder provides instant, cheap context.

Context Window Strategy: Planning vs. Building



Split your sessions. The `apply` command does not need the chat history of the planning phase.

Cost Analysis & ROI

Cloud Invoice

Cloud Code / Cursor Pro Max
\$100.00 / mo

Developer Capacity
1 Full-time or 2 Shared

Target Budget: \$100.00

ROI Benchmark: The 5-Task Challenge.

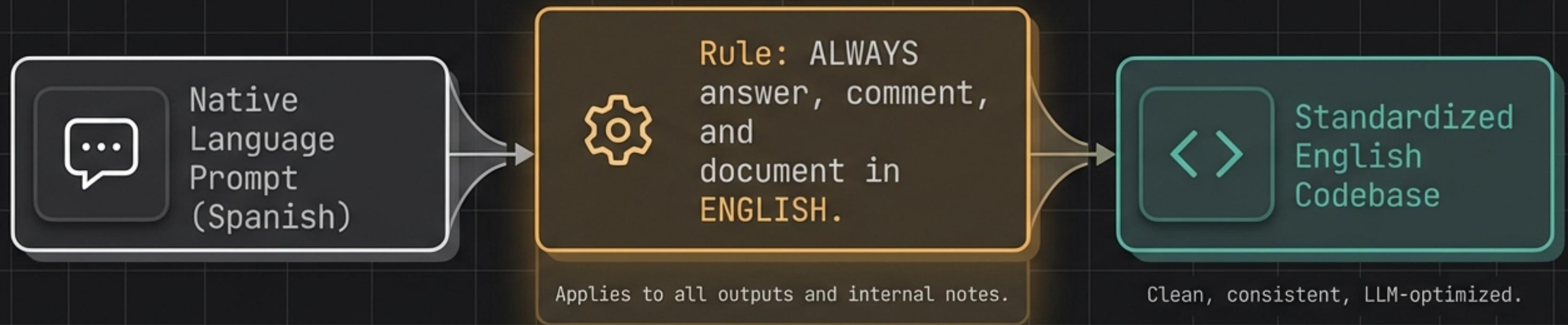
1. Create
2. Edit
3. Delete
4. View
5. Update

Completed in 2 afternoons. License pays for itself by minute three.

⚠ Warning: Hitting token limits implies a poor workflow (context pollution), not an insufficient license tier.

The Language Protocol

Native thinking, Global code.



Why? Ensures global standards and maximizes LLM performance on English-centric code patterns.

Integrating into Legacy (Brownfield) Projects



```
user@project:~$ spec init
> Initializing Open Spec Framework...
> [?] Select your tools:
✓ [x] Cursor
  [ ] VS Code
✓ [x] Cloud Code
> Structure generated.
> Structure generated.
```

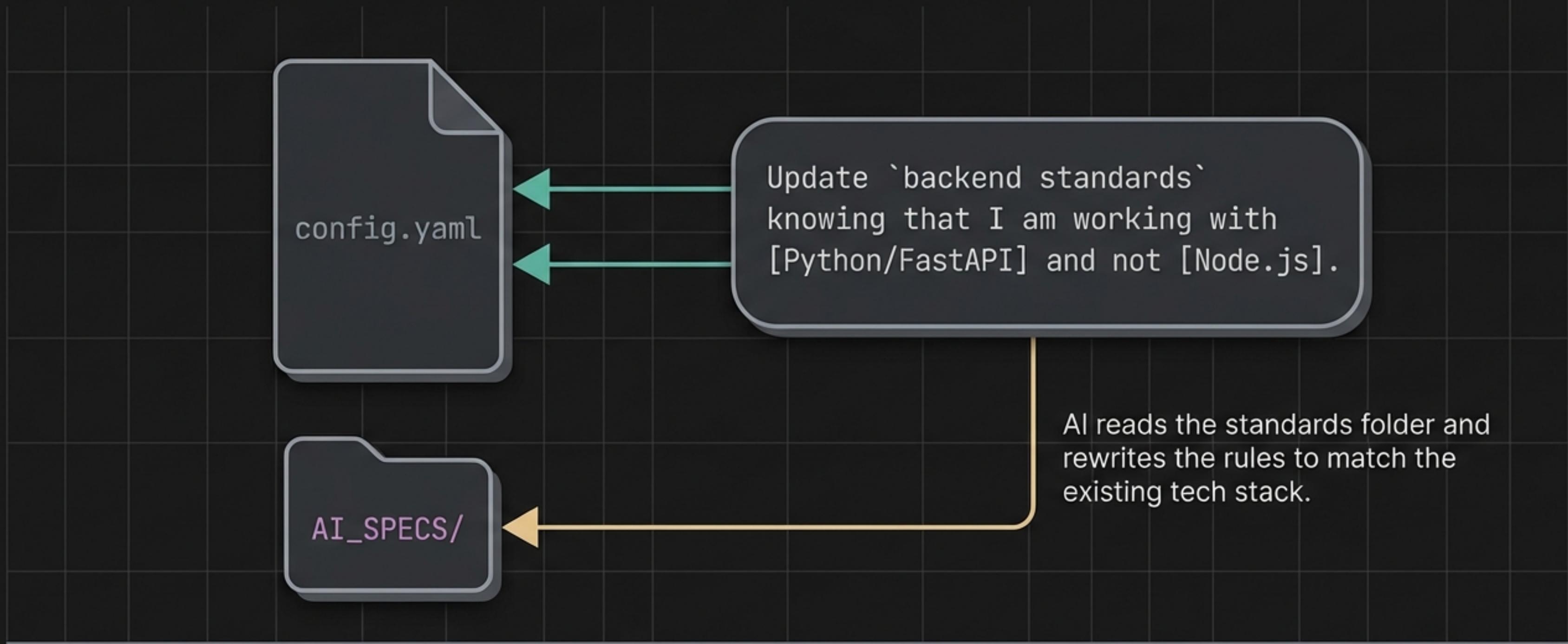
⚠ ACTION REQUIRED: Restart IDE to load commands.

Retrofitting Strategy:

1. Run initialization in root.
2. Select tools.
3. Restart.

Result: The `specs` folder starts empty. You do not rewrite the app; you initialize the framework on top of it.

Contextualizing the Configuration



The 'Secret Sauce': Explicitly telling the framework to adapt its default pattern to your specific legacy language.

The Command Stack

From Chatting to Commanding.

/new

Start Feature

/ff

Fast Forward (Plan & Spec)

/commit

Save State

/apply

Execute & Sync

Reduce AI 'chat' in favor of deterministic commands. It makes the workflow predictable, debuggable, and scalable.

The Standard for High-Velocity Devs

- Specs are the Source of Truth:
Code follows docs.
- Split Windows: Separate Planning
from Implementation.
- Enforce English: Standardize output
globally.
- Automate the Boring: Let the
framework handle the delta.

Next Steps:
The 5-Task Challenge

- Create Feature
- Edit Feature
- Delete Feature
- View Feature
- Update Feature

Download the Open Spec repo.

Documentation is no longer a chore; it is the engine that drives the code.