



A code repository built for AI.

A graph-native code repository for AI agents
and the people who build with them.

AI is backing teams **into a corner.**

AI tools generate more code. Teams still have to understand it, check it and keep it running.

 Review

+442%

median time
in review

Faros, April 2026

 Unreviewed

+31%

pull requests merged
with no review

Faros, April 2026

 Production

+125%

monthly production
incidents

Faros, September 2026

 The bill

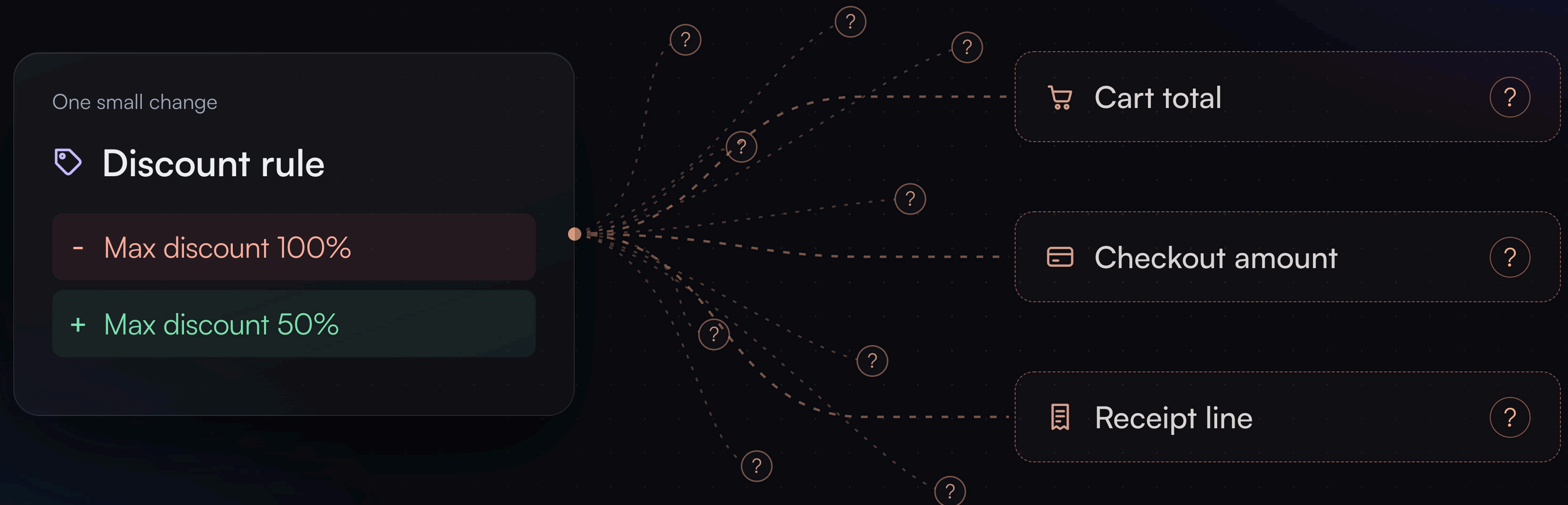
39%

of enterprises report
higher-than-expected
AI coding costs

Mavrik, July 2026

We keep paying to rediscover how code connects.

Without a shared record, the next person or agent may have to trace the same connections again.

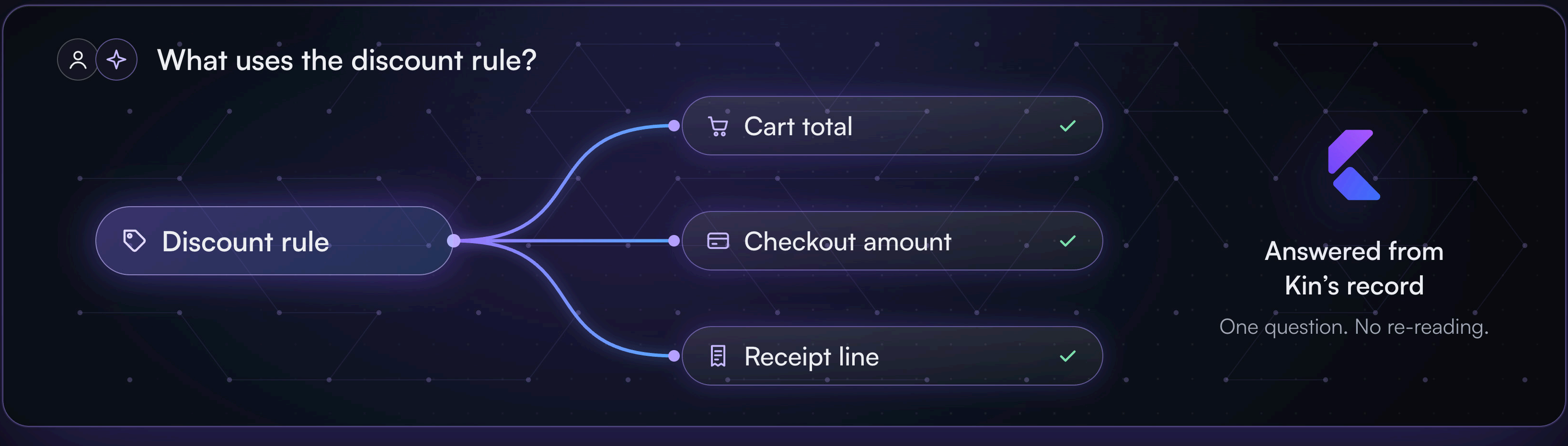


76% of tokens went to reading in one coding-agent study.
Claude Sonnet 4.5 on SWE-bench Verified.

Illustration based on Kin's saved discount example, September 16, 2026. Dashed paths mark connections to check. SWE-Pruner, 2026: read operations were 76.1% of tokens in this study, not a measure of avoidable reading or human review time.

Kin versions code and its connections so people and AI can reuse the answers.

The discount rule's recorded connections answer the same question at the next handoff.



~~+442% time in review~~

✓ **Faster review**

~~+31% unreviewed changes~~

✓ **More code checked**

~~+125% monthly incidents~~

✓ **Earlier risk checks**

~~39% report AI cost overruns~~

✓ **Less repeat reading**

From Kin's saved example of September 16, 2026, on Kin 0.7.19. Kin can miss connections. The bottom row is what Kin is designed to improve; the benchmark on the next slide measured tokens.

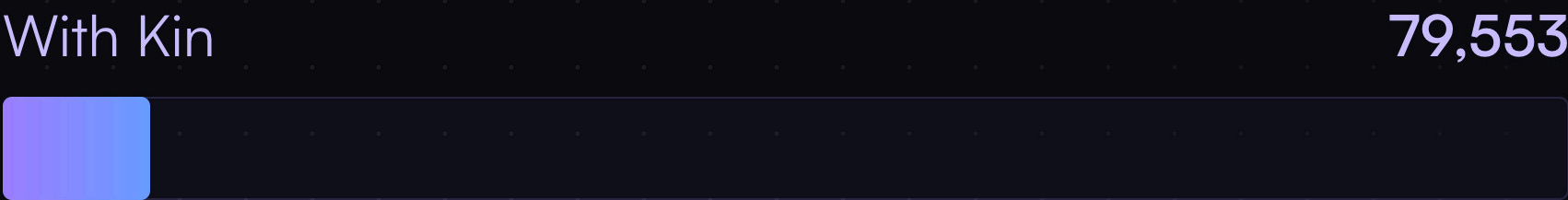
Kin answered the same questions with 91% fewer tokens.

A sealed test on held-out questions in React and Flask. Kin supplies the map, so AI can focus on the work.

91%

fewer tokens

Total prompt + completion tokens



79%

Correct on the first response.
No follow-up query.
19 of 24 runs

0

LLM tokens for Kin's lookup.
The graph supplies the answer.
The AI response still used tokens.

8 sealed questions × 3 seeds = 24 paired runs; all 24 Kin final answers scored correct. React + Flask; same Qwen3.8-27B model. Read-only, unreleased answer-first build, tuned earlier and frozen. Kin's answer entered the first prompt. Uncached tokens; no speed or customer-bill result.

A team of 200 developers could gain ~\$750K in annual value.

Illustrative annual scenario for a team using AI, before KinLab fees.



As AI use grows, so can the value.

Beyond the estimate: fewer mistakes, stronger customer trust and retention, and more time to deliver value.

Annual assumptions for 200 developers. Review: 6 h/week × 46 weeks × 10% saved. Rework: 10 fixes/month × 4 h. Incidents: 1/month × 40 h. Labor: \$94.47/h. AI: \$300/dev/month in variable usage × 19% assumed savings. Hours do not overlap. Sources: Bosu and Carver, BLS, Anthropic, LinearB. All reductions are hypothetical.

Kin is free.

KinLab is the paid team platform.

The first core pricing proposal is a subscription per developer seat.



Free

The open foundation

Run it yourself.

For individual developers and teams.

Your code, context and history.



Proposed core plan

\$40

per developer seat
per month

The managed team platform

Shared code context

Coordinated reviews and releases

Team and agent access controls

Potential expansion: hosted agents and automation usage, optional add-ons and enterprise packages.

Pricing hypothesis. KinLab is in development. \$40 per developer seat/month is the proposed core subscription; packaging, included usage and willingness to pay require validation. Expansion is planned and separately priced. Related models: Cursor, Devin and CodeRabbit.

Kin can grow from developer use or a leadership mandate.

First customers: teams of 20 to 200 developers building with AI agents on large codebases.





FOUNDER

Troy Fortin Jr

Founder and creator of Kin. Full-time on Kin when the round closes.

I built Kin after spending **my own money** on AI that kept **losing context** and sending me back to **fix its work**.

Hands-on experience

AI engineering at The Home Depot

Built from that experience, self-funded, by directing coding agents

Kin, in public beta

123

releases since June

14

languages supported

17,600+

automated tests

linkedin.com/in/troy-fortin-jr

github.com/firelock-ai/kin

Kin puts code connections inside the repository's own history.

The map becomes part of the foundation people and agents build on.

SELECTED EXISTING APPROACHES

Extend the existing repository model.

Each adds a layer on top of it.

Code hosts



GitHub



GitLab

Context engines



Sourcegraph



Augment

Agent records



Entire



Agent Trace

Supported by Cognition

We believe more layers are a costly, short-term answer.



Kin

Code and recorded connections,
versioned together.

Tracked content

Recorded connections

One repository history

Commit, branch and merge them together, entity by entity.

Patent pending

A versioned, tamper-evident graph of code
and its connections.

Stays current

Edits update only the connections they touch.
Kin flags anything not yet rechecked.

Works beside Git

Kin reads Git history without changing it,
so teams adopt it one repository at a time.

Our first year after funding moves from hiring to paying teams.

Milestones and targets for the first four quarters after the seed round closes.



What we raise on next: repeat team use, pilots that become production teams, and a repeatable way to win them.

A \$3.75M seed takes Kin from public beta to paying teams.

Spending follows what each milestone proves.

SEED ROUND

\$3.75M

24 months of runway

Valuation and instrument in conversation.



Superintelligence needs a better foundation.