

Cut your AI coding bill by 30–60%.

Without changing how your team works.

Same dev tools. No new harnesses. One-time, 5-minute setup.

NO BEHAVIOR CHANGE FOR DEVELOPERS

One-time install

brew install codelexica/tap/cdlx, then cdlx init.
About 5 minutes per developer, once.

Same tools, same workflow

Claude Code and Codex keep working exactly as before. No new IDE, no new agent, nothing to learn.

Savings you can verify

Every number is measured against your own team's usage, and a paired A/B test proves it on your tasks before you commit.

OPTIMIZE

Works with Claude Code and Codex. Other tools available on request.

Cut token spend 30–60%. Keep the quality.

Optimize runs as a lightweight sidecar next to each developer's coding agent. It steers model, effort, and context decisions in the background, so the same engineers ship the same work on far fewer tokens.

OPTIMIZE (STANDARD)

Without access to your codebase

up to 30% typical token-spend reduction

Meaningful savings from day one, with nothing indexed and no code leaving your machines.

Model routing. Each session runs on the cheapest model that can actually do the job, decided from task difficulty, not habit.

Effort steering. Reasoning effort is matched to the task, so routine work stops paying for deep-thinking budgets.

Subagent model pinning. Helper agents run on efficient models while the lead agent keeps the frontier model.

Session budgets. Per-session token budgets with org-level defaults, so a runaway loop cannot burn a day's spend.

Context management. Constraint narrowing, structured handoffs, and cross-provider continuity keep sessions lean without dropping what the agent needs.

OPTIMIZE (PREMIUM)

With your codebase indexed on Code Lexica

30–60% typical token-spend reduction

Everything on the left, plus persistent knowledge of your architecture and conventions, so agents stop paying to rediscover it on every run.

Advanced routing. Cheaper models succeed far more often when they are handed the right context, so more work moves down-market safely.

Just-in-time context. The architecture, conventions, and exact files that matter are injected automatically at session start and per task. No prompting, no copy-paste.

Agent-ready specifications. Task specs grounded in your codebase, written for the agent, so it starts implementing instead of exploring.

Persistent codebase knowledge. A living model of your repositories that every agent session draws on, kept current as the code changes.

PROOF

Two ways to prove the savings

ONE-OFF · CONTROLLED

Paired A/B test `cdlx ab-test run --task "..."`

A controlled experiment on your own repository, finished in an afternoon.

Paired worktrees. Same task, same base commit, in isolated worktrees: one arm with Code Lexica, one without.

Fair pricing. Both arms priced from one rate table, so API-key and subscription usage compare fairly.

Full scorecard. Tokens, cost, duration, verify-command pass rate, and a blind-judged quality score per arm.

Day-one results. Measured savings land on your Optimize dashboard before any baseline history exists.

ONGOING · REAL-WORLD

Ongoing Savings Tracker SAMPLE DATA

Live savings on your team's real work, measured against your own pre-Code-Lexica baseline.

Est. savings (API-rate value)
\$3.2K–\$4.1K
vs your team's usage before Code Lexica

Tokens / message
–31%
1,250 with · 1,800 without

Code Lexica adoption
9/12
devs using Code Lexica this window

Model mix · with Code Lexica

claude-sonnet-5 70%
claude-haiku-4-5 20%
claude-fable-5 10%

Model mix · without

claude-sonnet-5 50%
claude-fable-5 40%
claude-haiku-4-5 10%

Cost waterfall · \$16.2K → \$13.8K

Volume		+\$1.8K
Model mix		–\$1.1K
Input tokens / msg		–\$1.6K
Output tokens / msg		–\$0.9K
Cache		–\$0.6K



HOW IT WORKS

5-Minute One-Time Setup

1 Install

```
brew install codelexica/tap/cdlx
```

Install the Code Lexica CLI tool via Homebrew.

2 Track

```
cdlx init
```

Connect to your Code Lexica account and select repos to track.

3 Optimize

```
cdlx sidecar enable
```

Enable token optimization to run in the background.

4 Measure

```
cdlx ab-test run
```

Measure real savings live on actual coding tasks in your codebase.

TRACK

Know what your team is actually spending

Track meters real AI-coding usage across the tools your team already uses, for your whole organization.

- Agentic monitoring of your ongoing coding usage and spend
- Breakdowns by member, model, repository, and tool, updated automatically
- Weekly and monthly briefings and trend analyses
- Org-wide token usage leaderboards

TYPICAL ROLLOUT

From pilot to steady state

Pilot DAYS

One team installs the CLI and runs Track. A paired A/B test on your own repository gives you a measured number before anything is rolled out.

Rollout WEEKS

Optimize is offered to every developer as the last step of init. As coverage grows, the dashboard compares each developer against their own pre-Code-Lexica usage.

Steady state ONGOING

Org policy sets routing, effort, and budget defaults centrally. Savings are reviewed monthly against a baseline that keeps learning from your usage.

YOUR DATA

What leaves your machines

Sent to Code Lexica

- Token counts and model identifiers
- Tool-call names and result status
- Session timings and repository fingerprints
- Optimize intervention counters

Never stored

- Prompt or response text
- Tool arguments
- File contents
- Project paths (hashed by default)

Indexing your codebase for the second Optimize tier is a separate, explicit step. A/B-test task descriptions are stored because you write them for the test.

SOC 2 Type 1 certified; security program aligned to NIST 800-53 Rev 5. SSO/SAML, role-based access, encryption at rest. Details at codelexica.com/security.

If you don't save, you don't pay

Optimize is priced against the savings it creates for your organization. If we can't save you money, we'll refund you. Start a free 21-day trial today to see the savings yourself.

Start now (free 21-day trial)

codelexica.com/auth/signup
`brew install codelexica/tap/cdlx`

Book a Demo

hello@codelexica.com

We'll walk you through Optimize setup and show results same day.

