



How developers integrate route optimization

How the people who build or integrate the solution themselves handle route optimization today: practices, pain points, languages and tools, and what they expect from an API.



Who we surveyed, and how to read these figures

Kardinal surveyed 145 professionals, mostly from logistics and software publishing, about route optimization: how they handle it, where it breaks down, the languages and tools they use, and what they expect from an API. This report focuses on the people who build or integrate the solution themselves.

99

technical respondents

145

total respondents

15

questions covered

Summer 2026

Jul 1 → Sep 8

Key takeaways

The main findings, with the standout figures. The question-by-question detail follows.

56%**The problem is the field**

The gap with real-world conditions is the #1 difficulty (56%), ahead of performance / computation time (47%) and shifting business requirements (43%).

45%**Built in-house, at a cost**

Self-hosting remains the preferred model (45%) and in-house development dominates (56%). Yet among those who run their own stack, half report regular or heavy maintenance.

1 in 3**AI agents are already here**

One in three respondents uses an agent in their daily workflow, and only 17% use none. Claude Code is by far the most cited.

≈ 2 in 3**Public pricing matters**

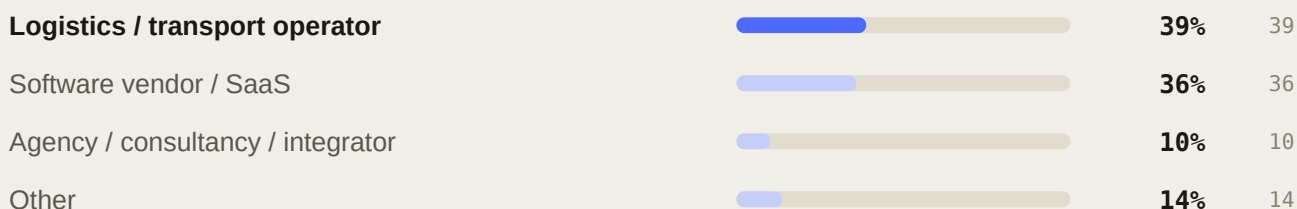
Nearly two thirds are strongly put off by, or refuse to even evaluate, an API without public pricing. Documentation and code examples also come first.

Who responded

Organization type, role, experience and involvement in route optimization.

Q1 Which type of organization do you work for?

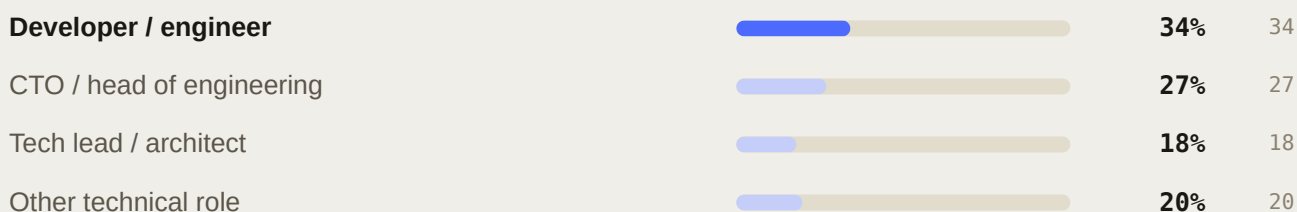
Single choice · base 99



→ The technical sample is split almost evenly between **logistics operators** and **software vendors**, ahead of agencies and other organizations.

Q2 What is your primary role?

Single choice · base 99



→ **Developers and engineers** make up the largest group, followed by CTOs and tech leads.

Q3 Have you ever worked on route optimization?

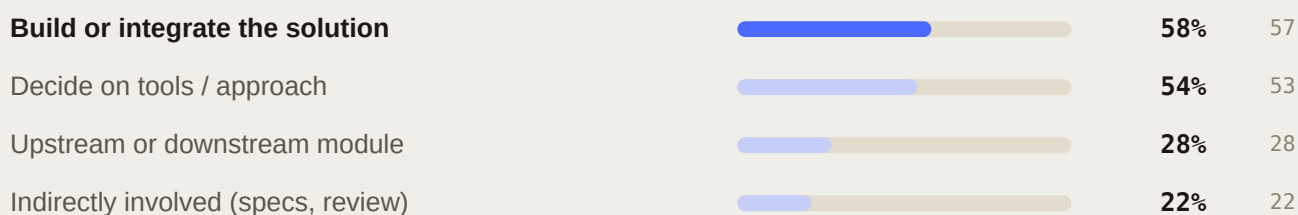
Single choice · base 99



→ The vast majority work on the topic **regularly or occasionally**.

Q4 How are you involved in route optimization?

Multiple choice · base 99



→ A majority **build or integrate the solution themselves**, and more than half decide on the tools or approach.

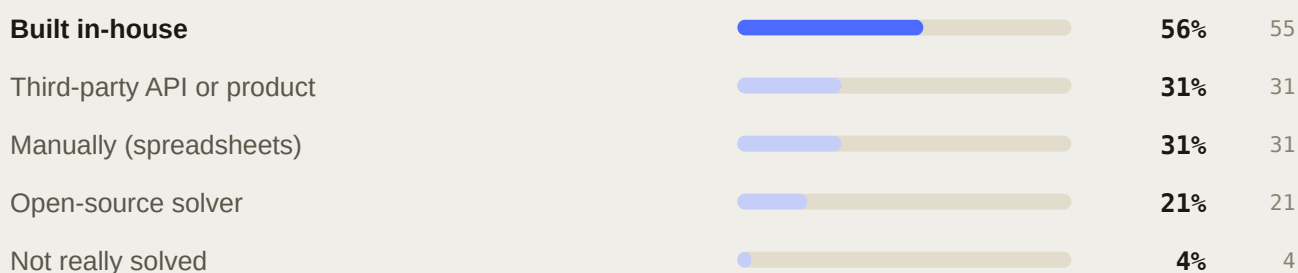
A relevant, highly qualified sample: technical experts, three quarters of whom work on route optimization hands-on, choosing the tools or integrating the solutions, and who therefore know the subject and its challenges inside out.

How the problem is handled, and where it breaks down

How route optimization is handled today, the difficulties teams run into, and the maintenance burden.

Q5 How is the problem handled today?

Multiple choice · base 99



→ **In-house development dominates**; third-party APIs and manual handling are tied, ahead of open source. One third of respondents (33%) combine at least two approaches; among those who build in-house, nearly half also rely on another approach.

Q5 Who combines what: one third use several approaches

Co-occurrence of approaches · read by row

USES...	+ IN-HOUSE	+ THIRD-PARTY	+ MANUAL	+ OPEN-SRC	ONLY ONE
Built in-house (55)	—	20%	24%	18%	53%
Third-party API or product (31)	35%	—	19%	26%	48%
Manual (31)	42%	19%	—	16%	42%
Open source (21)	48%	38%	24%	—	33%

→ Read by row: among those who **build in-house**, 53% use nothing else. The most hybrid group is **open source**: only a third rely on it alone.

Q6 Which tool(s) or library(ies) specifically?

Open-ended · 37 responses

Three families of tools stand out, listed here by number of mentions:

■ Open-source solvers

1. OR-Tools
2. VROOM
3. Timefold
4. pgRouting

■ Mapping & routing

1. Google Maps
2. ESRI / ArcGIS
3. GraphHopper
4. OpenRouteService
5. OSRM
6. HERE

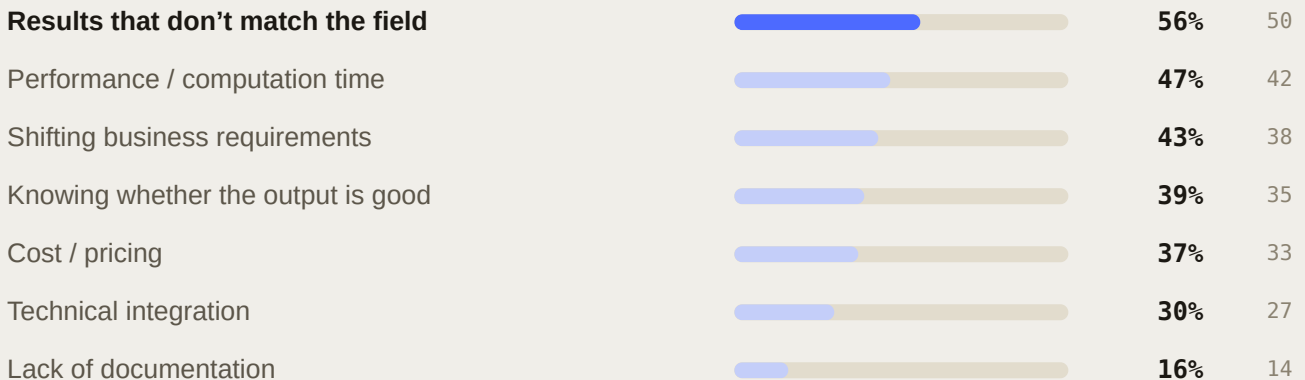
■ Platforms / TMS

1. Kardinal*
2. Nomadia
3. Ortec
4. PTV
5. Urbantz

* As this survey was distributed by Kardinal, its customers may be overrepresented among respondents.

Q7 What has been the hardest part?

Multiple choice · base 89



→ The **gap with real-world conditions** clearly dominates, ahead of performance and shifting business requirements.

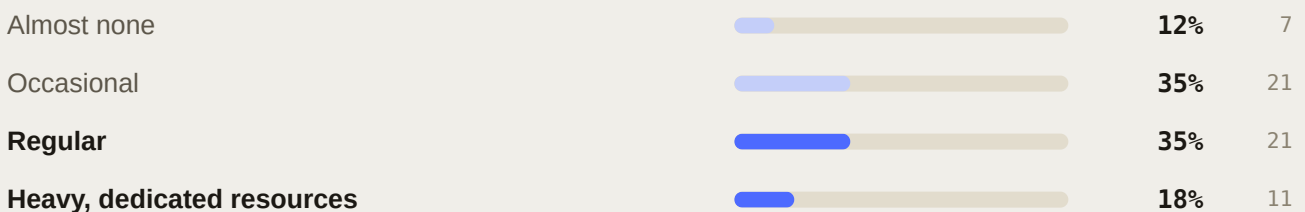
“

The main blind spot of current solutions is field validation: it is hard to know whether an optimized route is actually better than one built empirically by an experienced driver. Metrics measuring the gap with reality would be more useful than purely algorithmic ones.

A respondent, open-ended answer (translated from French)

Q8 How much maintenance does your stack require?

Single choice · conditional · base 60 · runs its own stack



→ Asked only of those who run their own stack. The burden remains significant: **half describe it as regular or heavy**.

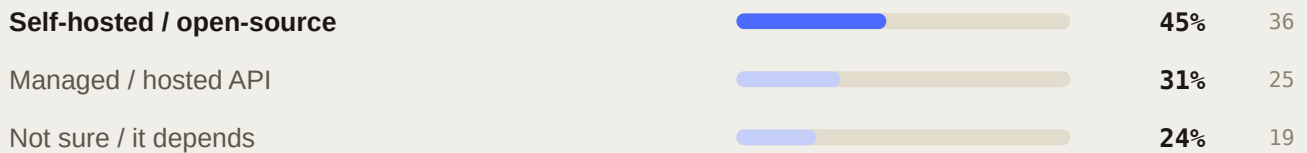
A real pain point. Most teams build their own solution (56%), in nearly half of cases alongside another approach, even though half of those running their own stack report regular or heavy maintenance. Difficulties affect a large share of respondents (up to 56%), whatever the approach: route optimization remains a complex problem, poorly served by existing solutions, whether in-house, open-source or third-party APIs.

What development teams prefer

The preferred deployment model, the languages used, and what matters when adopting an API.

Q9 Which model do you lean toward?

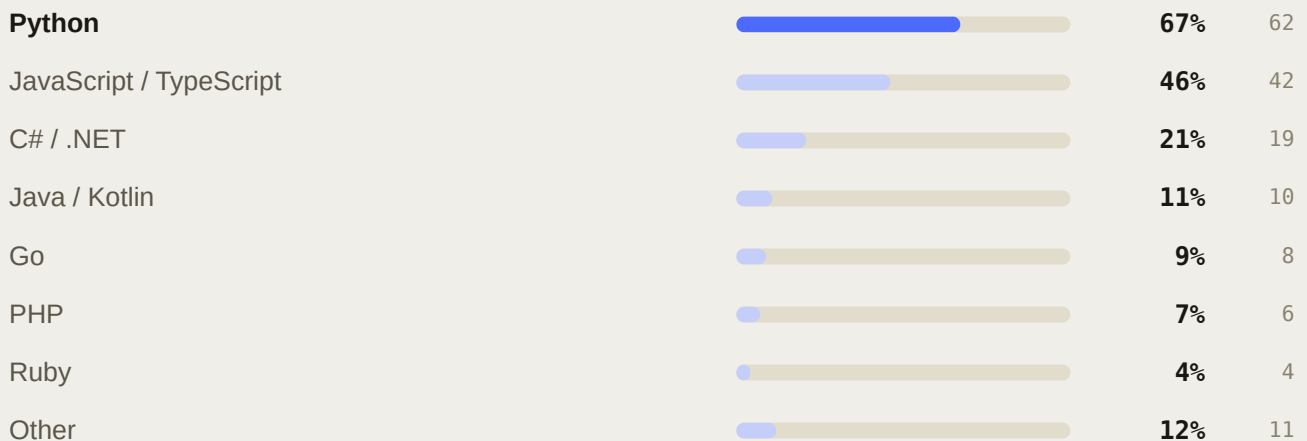
Single choice · base 80



→ Self-hosting remains the preferred model, ahead of managed APIs, with a notable share of undecided respondents.

Q10 Which languages do you use?

Multiple choice · base 92



→ Python clearly dominates, followed by JavaScript / TypeScript.

Q11 When adopting an API, what matters most?

Forced ranking of 7 items · base 92 · sorted by ascending average rank

CRITERION	AVG. RANK	RANKED 1ST	TOP 3
1 Clear, complete documentation	3.37	26%	59%
2 Code examples / templates	3.60	14%	58%
3 Guidance on modeling	3.85	18%	39%
4 Sandbox to test without commitment	3.98	7%	46%
5 Predictable, transparent pricing	3.99	18%	39%
6 Well-designed SDK in my language	4.50	10%	35%
7 Fast time to first call	4.72	7%	25%

→ **Documentation and code examples** lead the adoption criteria; the ranking remains fairly flat (average ranks are close together).

Choosing an API comes down, first and foremost, to essential reassurance: clear documentation, code examples, guidance on modeling one's problem. In other words: is the API genuinely built for developers?

The role of AI agents

Actual use of coding agents and the tools cited.

Q12 How present are AI agents in your workflow?

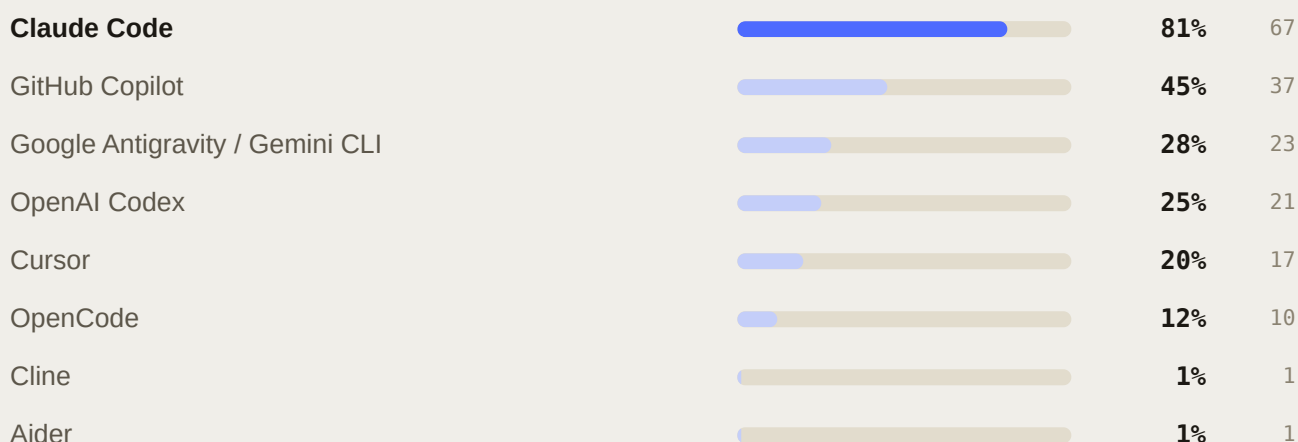
Single choice · base 92

Not at all		17%	16
Experimented, nothing in production		20%	18
Occasionally		23%	21
Part of the daily workflow		35%	32
Products where agents call APIs		5%	5

→ AI agents are well established: **one in three uses them daily**, and fewer than one in five uses none at all.

Q13 Which coding agents do you use?

Multiple choice · base 83 · cited at least one tool



→ Among agent users, **Claude Code is by far the most cited**. 65% cite at least two tools, 2.1 on average.

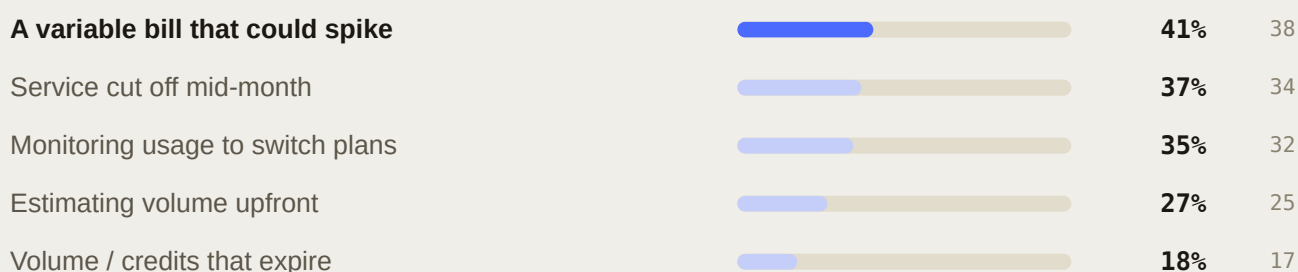
AI agents are becoming the norm. Measured in summer 2026, these figures are probably already out of date.

Pricing and hidden prices

Pricing pain points and reactions to an API without public pricing.

Q14 Which aspects of pricing bother you most?

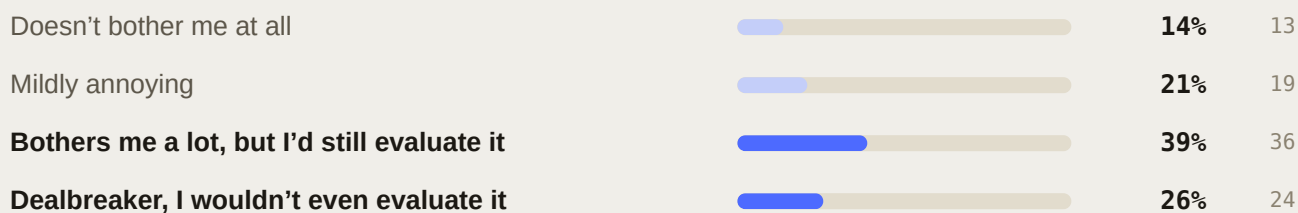
Multiple choice · base 92



→ An **unpredictable bill** comes first, ahead of service being cut off mid-month and having to monitor usage.

Q15 An API without public pricing: how much does it bother you?

Single choice · base 92



→ Nearly **two thirds** are strongly put off by, or refuse to even evaluate, an API without public pricing.

The market expects true self-service at a manageable cost. Documentation and examples come first when adopting an API, nearly two thirds are strongly put off by an API without public pricing, and an unpredictable bill is the top pain point. This is consistent with the fact that nearly half favor self-hosted / open-source (45%), which on paper ticks both boxes.

What the cross-analysis reveals

Insights that only emerge when answers are cross-referenced.

01 Building it yourself doesn't solve the field problem

The gap with real-world conditions affects in-house teams as much as everyone else (55% vs. 58%): building it yourself doesn't remove the #1 difficulty, and it exposes teams more to performance issues (53% vs. 37%) and integration issues (35% vs. 24%).

02 Claude Code almost always comes with another tool.

76% of Claude Code users also cite another tool (2.4 on average): Copilot (40%), Gemini (30%), Codex (28%), Cursor (22%). Multi-tooling is the norm.

03 Each sector has its own language signature.

Python dominates everywhere (logistics 56%, software vendors 70%), but C# / .NET is a hallmark of logistics (33% vs. 12%), while software vendors are twice as likely to use JS / TS (64% vs. 31%).

04 Sensitivity to hidden pricing is not uniform.

Software vendors are the most put off by an API without public pricing (76% vs. 53% in logistics). Logistics is more sensitive to estimating volume upfront (28% vs. 18%), software vendors to monitoring their usage (45% vs. 25%).

An API built on these findings

These results directly shape how we are building our route optimization API.

THE FINDING

The gap with real-world conditions is the top difficulty (56%), ahead of performance and constantly shifting business requirements.

OUR RESPONSE

The real-world gap mostly comes from actual constraints the model cannot represent. Our engine covers an unusually large share of them: 93% of a reference set of 149 business constraints across 8 sectors, while none of the ten other engines compared reaches 75% (August 2026 benchmark). We also publish a guide and examples drawn from our real-world cases.

THE FINDING

Documentation and code examples matter most when adopting an API, and nearly two thirds are put off by an API without public pricing.

OUR RESPONSE

A documented REST API, with explicit input / output schemas, runnable request examples and a public price list. Everything you need to read the specs, test a request and cost out an integration without talking to sales.

THE FINDING

The main pricing pain points: an unpredictable bill that can spike, and the risk of having the service cut off mid-month.

OUR RESPONSE

Pricing indexed to a clear, measurable unit: the number of unique stops per 24-hour window. What happens beyond your plan is defined upfront: no service cut-off mid-month.

THE FINDING

One third of respondents already run AI agents daily, and many struggle to judge whether an optimization result is actually good.

OUR RESPONSE

A programmatically callable API that returns structured outputs (JSON), along with indicators on solution quality and on which constraints are met or violated. An agent can judge a result and move on to the next action.

THE FINDING

Many build in-house or self-host, but maintaining that stack remains regular or heavy work for a good half.

OUR RESPONSE

We run optimization as a managed SaaS: you call the API, and we take care of the solver, the tuning and algorithm updates. Maintaining the optimization layer no longer falls on your teams.

Survey "How developers integrate route optimization" · technical profiles (99 of 145) · collected July 1 – September 8, 2026 · published September 2026 · self-reported data, self-selected sample. API figures and scope: Kardinal press release of September 28, 2026.

The API, open to all

Everything you need to go from the survey to your first optimization.

Test the API on your own routes

Clear documentation, input / output schemas, runnable request examples and a public price list. Create an account and make your first call, no sales call required.

[Read the documentation](#)[Open the console](#)

`developers.kardinal.ai` · `console.kardinal.ai`

1,000

optimized stops
free

from €0.012

per optimized stop

1× / 24 h

each stop billed
only once

Self-service

no sales
contact

What's included

- ✓ Python SDK and developer console
- ✓ Documentation designed for machine reading
- ✓ Runnable request examples
- ✓ API keys and usage tracking
- ✓ Modeling guides based on real industrial cases
- ✓ Public, transparent pricing



Ten years of logistics optimization, now an API call away.

Since 2015, Kardinal has been developing route optimization technology for transport and logistics companies, combining operations research and machine learning to model the most complex business constraints.

After supporting carriers and shippers through custom deployments, Kardinal is now opening up the same expertise as a self-service API, designed to be used by technical teams and AI agents alike.

Recognized as a Representative Vendor in the Gartner Market Guide for Vehicle Routing and Scheduling 2025, renewed in 2026.

