opoint

You can build a news pipeline in an afternoon.

Keeping it alive is the hard part.

A build-vs-buy guide for product and data teams scoping a news data component.

Inside this guide

- Why the prototype is the easy part, and production is everything after.
- The maintenance costs nobody scopes: source upkeep, deduplication, multilingual coverage.
- The security reality of AI-generated ingestion code, with the data to back it.
- The silent coverage hole that breaks stories in local languages before the English wires move.
- A three-question test for when building really is the right call.
- What buying actually replaces, mapped cost by cost.

Estimated read time: about 9 minutes

SECTION 01

The demo solved the easy part. Production is the rest.

With today's AI tools, a working news-monitoring prototype takes an afternoon. You point it at a few sources, articles start flowing, the demo lands, and it feels like it's solved.

It isn't.

The prototype tested the easy part: a handful of known sources, one language, clean data on a quiet day.

Everything that makes a news pipeline hard shows up later, at scale, when a customer depends on the feed and a story you never saw has already broken.

This guide is for the team weighing that exact decision. If you're scoping a news data API, a news monitoring component, or a real-time news feed for your product, here's the honest breakdown of production costs and guidance on whether to own it or buy it.

SECTION 02

The build is a one-off. The coverage is forever.

Opoint monitors more than 250,000 sources. On a typical day, around 97,000 distinct domains publish something. That second number is the one that matters.

It is easy to accumulate URLs. Keeping them alive is much harder. Site redesigns can break your parser overnight. Feeds go dark. Paywalls appear. A source list isn't built once; it's maintained every week, usually after a customer spots the gap before you do.

And the work is judgement, not code. AI can write a flawless parser. It can't tell you which two hundred sources to point it at, or which fifty to drop when they no longer matter. That's the part no tool hands you.

The hidden costs, and when each one bites

HIDDEN COST	WHAT IT ACTUALLY MEANS
Source maintenance	Parsers break weekly as sites change. This is permanent, ongoing work, not a launch task.
Language expansion	Each new market is a fresh build: new sources, new encoding, new edge cases.
Deduplication	Invisible until volume arrives, then it becomes the difference between signal and noise.
Provenance	Source and timestamp on every item, defensible under audit. Your problem the moment a regulator asks.
Security	AI-generated ingestion code carries risk you can't see until someone exploits it.
Licensing	Scraped content carries copyright and terms-of-use exposure that surfaces at the worst time.

FOR THE ENGINEER IN THE ROOM

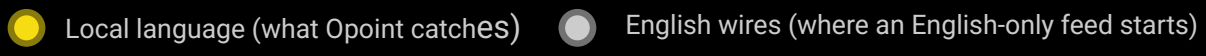
This is the section to forward internally. The maintenance burden is the part the commercial side consistently under-scopes. If you've maintained scrapers before, you already know the selector roulette, the silent feed failures, the 11pm pages.
Put a realistic weekly-hours number on it before anyone signs off on "we'll just build it."

SECTION 03

The coverage hole is silent by design

There's a difference between data arriving and data reaching the right decision point. A signal that lands in an inbox and requires manual assessment before anyone can act on it is likely to be missed when the inbox is full, the analyst is in a meeting, or the alert fires at 11pm.

The harder version of this problem is language. The stories that matter most often break first in a local-language outlet, hours or even days before the English-language wires move. An English-only feed doesn't just arrive late. It doesn't even know the story exists yet.



Vale mine overflow, Brazil — Portuguese

English wires 12h 25m behind

Estado de Minas · 16:05, 25 Jan

MT Newswire (EN) · 04:30, 26 Jan

Baogang explosion, Inner Mongolia — Chinese

+6h 02m head start

iFeng · 03:25, 18 Jan

Xinhua (EN) · 09:27, 18 Jan

Novi Sad station collapse, Serbia — Serbian

+1h 23m, and this is the small one

in4s.net · 12:34, 1 Nov

AFP / Barron's (EN) · 13:57, 1 Nov

This is the gap Opoint is built to close: 250,000+ sources across 135+ languages and 230 jurisdictions, curated by people whose job is keeping that list relevant.
An English-only feed starts the clock late. Local-language coverage starts it when the story actually breaks.

SECTION 04

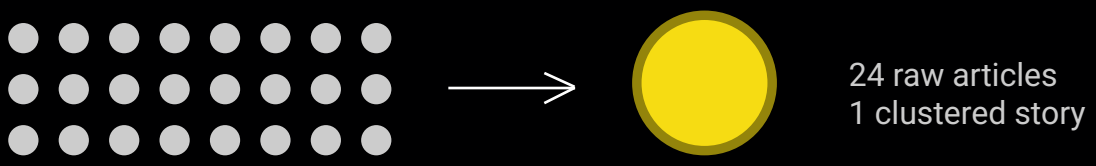
Volume is where pipelines drown

A single major event can generate hundreds of near-identical articles within an hour as a wire story is republished across outlets.
Treat each as separate, and you've built a noise machine: alerts that cry wolf, users buried, and the signal lost in its own echo.

Recognising duplicates from genuinely new developments, clustering related coverage, and surfacing the update that matters are among the hardest engineering challenges across the entire pipeline. News clustering is a data-stream problem whose cost grows non-linearly with volume, and teams routinely underestimate it because it's invisible in a demo.

It only appears at volume.
Which is to say, in production.
Which is to say, **after you've already committed.**

THE SHAPE OF IT



Getting that ratio right, reliably, at scale, in real time, is the work. Getting it wrong is how a monitoring product becomes the thing users mute.

SECTION 05

The risk that surfaces when someone's looking

If you build for compliance, risk, or financial services, two costs sharpen.

First, provenance. You need the source and timestamp for every item, defensible under audit. A feed with clean text but fuzzy sourcing is a liability waiting to surface, and it will surface at the worst possible moment. This is also why scraped content is a growing legal exposure, not just a technical one: the copyright and terms-of-use questions around scraping are now live, and licensed, traceable sourcing is something a scraped pipeline can't replicate by definition.

Second, the security reality of AI-generated code. This is the cost that has changed most over the past year, and it's the one almost nobody scopes.

What the research actually shows

- On a 200-task benchmark built from real GitHub projects, the best-performing agent configuration were **61% functionally correct, but only 10.5% were both correct and secure.** Over 80% of the functionally correct solutions still contained a vulnerability (SusVibes benchmark, Carnegie Mellon et al., 2025).
- A scan of 5,600 publicly deployed vibe-coded applications found **2,000+ high-impact vulnerabilities, 400+ exposed secrets and 175 cases of exposed personal data**, all live in production (Escape.tech, October 2025).
- Separately, a study of 100+ models found that AI **chose the insecure option in 45% of code-generation tasks**, with Java failing in more than 70% of cases (Veracode GenAI Code Security Report, 2025).

A working prototype can still carry a liability you can't see, especially once it goes into a product handling sensitive data for regulated customers.
Functionally correct is not the same as safe to ship.

FOR THE ENGINEER IN THE ROOM

These figures describe agentic, best-case setups on public benchmarks, not worst-case cherry-picks. That's the point. Even at the strong end of the range, most AI-generated code that runs is not code that survives a security review. For an ingestion layer touching third-party content and customer data, that's the risk surface you'd be signing up to own and patch, indefinitely.

The costs you discover in year two

The build-vs-buy maths is usually done in year one.
The costs that actually decide it tend to land later.

01 **Licensing and copyright exposure.**
Scraped content sits on shifting ground regarding use and copyright. Licensed, traceable sourcing isn't a nice-to-have once a customer's compliance team starts asking where your data comes from.

02 **Switching and lock-in risk.**
Cheap developer-tier APIs often change their pricing or quotas once you depend on them. Teams that built on a free tier have been caught out before. The cost of being moved off a feed you didn't control is a year-two cost you pay at the worst possible time.

03 **The relevance-tuning grind.**
Raw coverage is the starting point. Months of deduplication, false-positive suppression and relevance tuning are what stand between "data arrives" and "data a customer trusts".

04 **Opportunity cost.**
Every engineer maintaining parsers is an engineer not building the product; only you can build. That's the real line item, and it compounds.

SECTION 07

When you should build: the three-question test

Sometimes building is the right call, and a guide that won't admit that isn't worth reading.

If your coverage needs are narrow, you're comfortable staying English-only, and your volume is low enough that duplication never becomes a problem, a handful of known sources will do. Then build it.
You retain full control, avoid ongoing data costs, and keep maintenance manageable.
For a tightly scoped internal tool, in-house is often exactly right.

ASK YOURSELF	IF THE HONEST ANSWER IS NO...
Is your coverage narrow and stable?	Source maintenance becomes a permanent tax that compounds.
Is English-only genuinely enough for your users?	You inherit the silent coverage hole, and you won't see what you're missing.
Is your volume low enough that duplicates never pile up?	Deduplication becomes the hardest part of your roadmap, in production.

Three yeses: build with confidence.

Any no: price the alternative properly before you commit, because that's where the maintenance tax starts compounding and the build quietly becomes the expensive option.

SECTION 08

What ‘buy’ gives you

If the three-question test pointed you towards buying, here’s what the bought version looks like, mapped to the costs this guide has walked through.

THE COST	WHAT BUYING REPLACES IT WITH
THE JUDGEMENT GAP	Human-curated, quality-checked coverage. The source list reflects editorial judgement, maintained by people whose job is keeping it relevant.
SOURCE MAINTENANCE	Coverage you don’t curate yourself: around 250,000 sources across 135 languages, maintained so you don’t inherit the source-maintenance job.
THE SILENT COVERAGE HOLE	Local-language coverage across 230 jurisdictions, so the story reaches you when it breaks, not when it reaches an English wire.
VOLUME / NOISE	Deduplication and clustering handled, so volume is the provider’s problem, not your 2am one.
PROVENANCE & AUDIT	Source, timestamp and the identifiers an audit needs, on every item
LATENCY	News indexed and delivered in minutes, not an hourly batch dressed up as real-time. Average delivery under 7 minutes from publication.

That’s the real build-versus-buy trade: not whether you can build a prototype, but whether you want to own all of this, forever, instead of the product only you can build.

SECTION 09

The numbers, like for like

If you're comparing providers or comparing a buy against your own build, these are the figures to hold everyone to.

250K+	3.5M+	135+	230	4-7 min
Sources monitored	Articles indexed daily	Languages covered	Jurisdictions	Delivery latency

Making the case internally

If you've read this far, you're likely the person who has to convince someone else. Here's the short version to paste into a deck or a message to your sponsor:

The prototype proves the concept. It doesn't prove production.
Building our own news pipeline means owning source maintenance forever, solving deduplication at volume, covering the languages our users actually need, defending provenance under audit, and patching AI-generated ingestion code that mostly fails security review.

Buying replaces all five with a maintained feed, so our engineers build the product only we can build. We should at least price the alternative before we commit to the build.

The prototype was the easy part

AI changed how cheap the first version looks. It didn't change what a production news pipeline requires: broad, well-chosen coverage, deduplication that holds at volume, multilingual handling, defensible provenance, security that survives scrutiny, and someone to keep all of it alive as the web keeps shifting.

If you've got a prototype running and you're weighing what production really takes, bring us your real markets, your real languages, and a real event you care about. We'll walk through what our data actually did with them. No demo theatre, just the numbers, so you can compare like for like.

You can also see our live source activity for yourself at **world.opoint.com** Sources appearing, disappearing and changing over even a few weeks gives a fair impression of what maintaining broad news coverage actually involves.

BOOK A DATA WALKTHROUGH

opoint

Legal Disclaimer

This guide is provided for informational purposes only and does not constitute legal, financial, or investment advice. The examples and opinions expressed are based on sources believed to be reliable as of publication, but no warranty is given as to accuracy or completeness. Organisations should consult their professional advisors when developing risk management strategies. Opoint and the author accept no liability for actions taken based on this document. All trademarks and company names mentioned are the property of their respective owners.

© 2026 Opoint.
All rights reserved.

Contact:
Email: marketing@opoint.com
Website: opoint.com

Connect:
LinkedIn