

Smarter Navigation For AI Coding Agents with Sonar Vortex

PRIMARY

Publisher	Sonar
Published	2026-09-01
Retrieved	2026-09-04
Cite key	sonar2026smarter
URL	https://www.sonarsource.com/blog/stop-the-context-tax/

Assessment. Vendor engineering blog by a named research engineer, publishing traces from Sonar's own development sessions. The measurements are self-reported and the post exists to introduce Sonar Vortex, so the product framing is interested. The session figures themselves are of Sonar's own work, not a competitor comparison, which is the part relied on here. The ~\$0.20/million cache-read price is stated without naming a model or provider and is carried as the source's own assumption.

Claims drawn from this source. CLAIM-2026-09-04-01, CLAIM-2026-09-04-02, CLAIM-2026-09-04-03, CLAIM-2026-09-04-04, CLAIM-2026-09-04-05

Published September 1, 2026. **Author** Antonio Aversa, Research engineer.

The context tax

"Read a 600-line file on turn 40 of a 512-turn session and you haven't paid for 600 lines. You've paid for 600 lines x ~470 more turns."

One pull request, traced

An 800-line backend change, single ticket:

- Model round-trips in the session: **512**
- Context window at its peak: **458,700 tokens**
- Cache-read tokens (the re-billed transcript): **152.8 million**
- Approx. cost of the session: **~\$41**

On a single unnecessary file read: "Re-billed as a cache read each turn: 5,770 x 470 ~=**2.7 million tokens** of pure waste." "At current cache-read pricing (~\$0.20 / million), that's **~\$0.54, for one unnecessary file read.**"

Across 18 comparable PRs

"Across **18** comparable single-ticket PRs in the same repo, the average was: **~234 million context tokens per PR**, **~\$65 per PR** (median ~\$52), **~700 model round-trips per PR**, context windows **routinely peaking between 450k and 975k tokens.**"

Sonar Vortex and SemSitter

Sonar Vortex is described as an enterprise harness for AI coding agents built on SemSitter, an in-house semantic navigation engine. Instead of grep-based file search, it maintains a Unified Dependency Graph (UDG) of the codebase in which nodes are functions, methods, classes, fields and parameters, connected by typed edges (calls, references, returns). An agent querying the graph for a code element receives that node plus its relevant relationships rather than whole files, which the post presents as the way to stop paying the context tax.

Methodology: the measurements come from real PR traces collected while building SemSitter with coding agents, with full session logging.