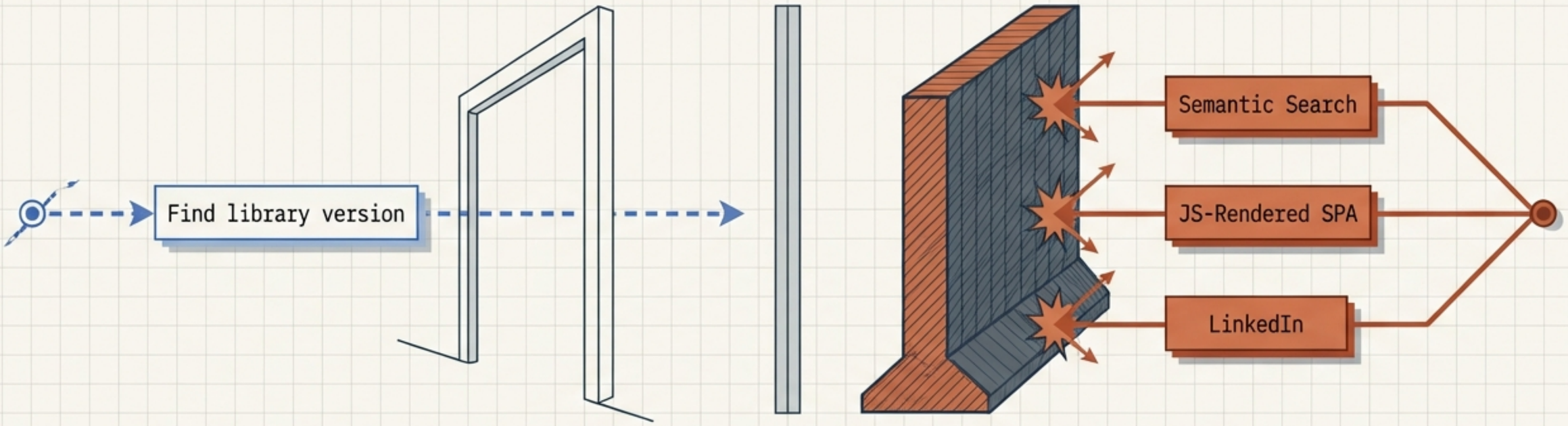
Hitting the Wall with Built-In Tools



SUCCESS: Simple queries pass through.

FAILURE: Complex needs are blocked.

WebSearch Limitations

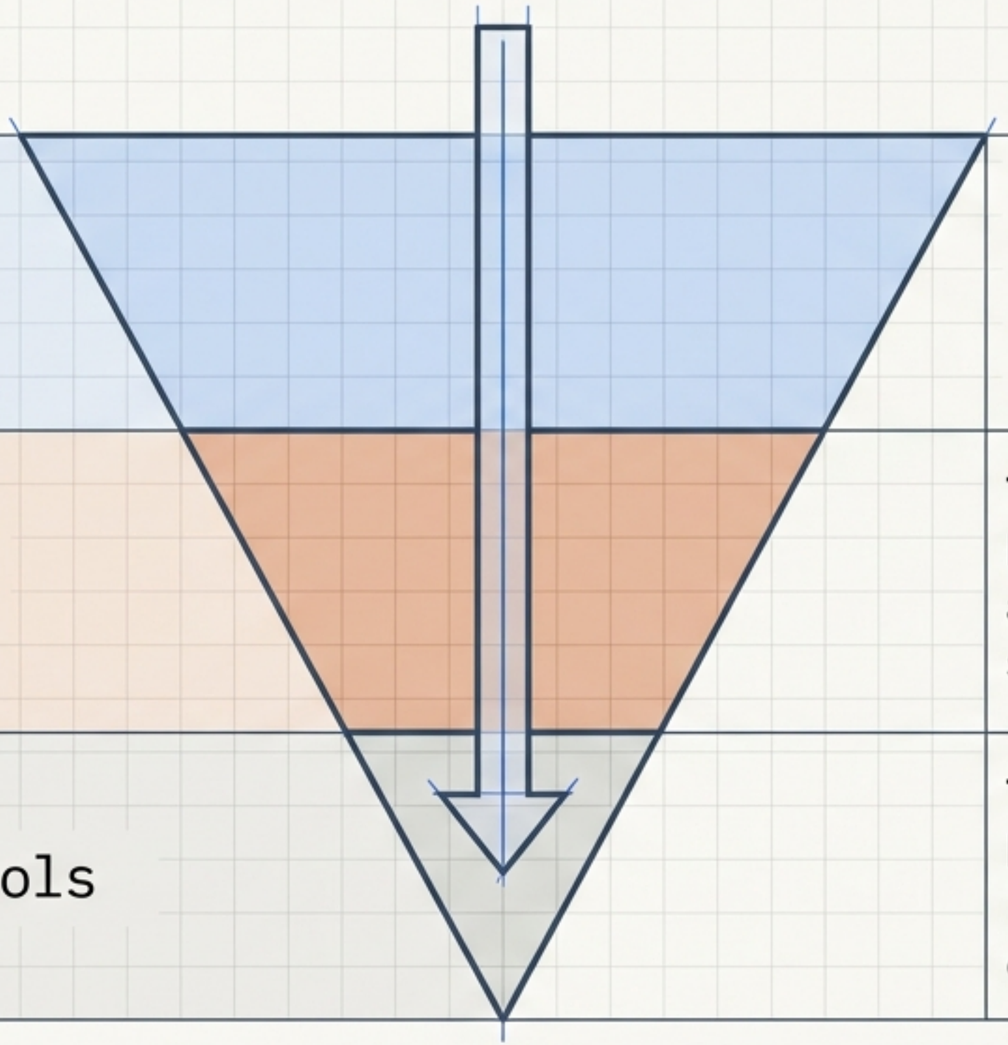
- Keyword matching only; zero semantic mapping.
- Lacks date filtering to exclude stale noise.
- Cannot scope results to specific domains.

WebFetch Limitations

- Cannot execute JavaScript (returns empty divs on modern SPAs).
- Entirely blocked by LinkedIn.
- Fails against standard anti-bot detection.

A Priority-Layered Routing Stack

Tier 1: Exa	**The Primary Researcher.** Semantic search, conceptual routing, date filtering, and GitHub/docs context.
Tier 2: Firecrawl	**The Heavy Lifter.** Full JavaScript rendering, anti-bot bypass, and structured site scraping.
Tier 3: Built-In Tools	**The Free Fallback.** Handles basic static page requests (WebSearch / WebFetch) to conserve API credits.



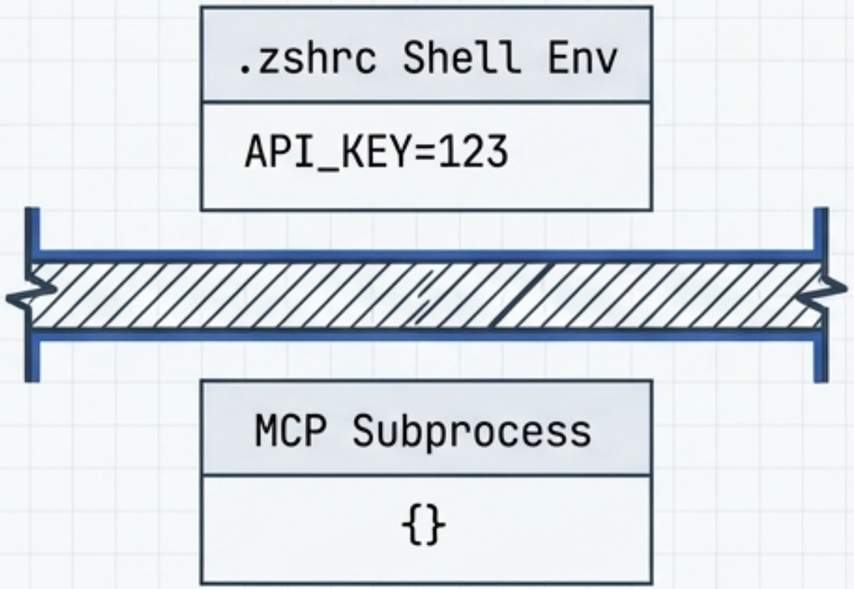
The Tool Capability Matrix

	WebSearch	WebFetch	Exa (MCP)	Firecrawl (MCP)
Semantic Search			✔	
JS Rendering				✔
Date Filtering			✔	
LinkedIn Access			✔	✔
Code/Docs Context			✔	
Structured JSON Crawling				✔

Exa and Firecrawl do not replace built-in tools; they architecturally fill the precise gaps left by keyword matching and static fetching.

The Secure Fix: Process Isolation via Wrapper Scripts

The Obstacle



MCP servers run as isolated child processes. They do not inherit shell exports.

Hardcoding keys into `~/.claude.json` creates version control security risks.

The Blueprint

Forces bash to export variables loaded from the gitignored .env file into the local environment.

```
#!/usr/bin/env bash

set -a
source $HOME/.claude/secrets/secrets.env
set +a

exec npx -y exa-mcp-server
```

Replaces the shell process with the MCP server, securing the keys in a clean process tree.

Orchestrating the Stack: The Workflow

1. Understand	2. Decompose	3. Parallelize	4. Deep Read	5. Synthesize	6. Save
Define the specific question and completion criteria.	Break primary goal into 3-5 independent sub-questions.	Launch Haiku-model agents simultaneously.	Extract full structured content from high-relevance URLs.	Compile findings, cross-reference, and enforce strict citations.	Output compiled markdown and update vector memory.

Parallel execution compresses a sequential 15-minute research task into 4 minutes of wall-clock time.

Production Smoke Tests

Exa Test Run

Query: 'MCP adoption 2026'

Filter: published > 2026-01-01

Result: Filtered dates successfully. Matched semantic concepts bypassing exact keyword constraints.

Firecrawl Test Run

Target: modelcontextprotocol.io

Action: firecrawl_scrape

Result: Extracted fully populated JS-rendered dynamic DOM tree as clean markdown. markdown. (1 credit)

Critical Implementation Note: The crawling_exa tool expects the urls parameter strictly as an array. Passing 'urls': 'https://…' silently fails. It requires 'urls': ['https://…'].