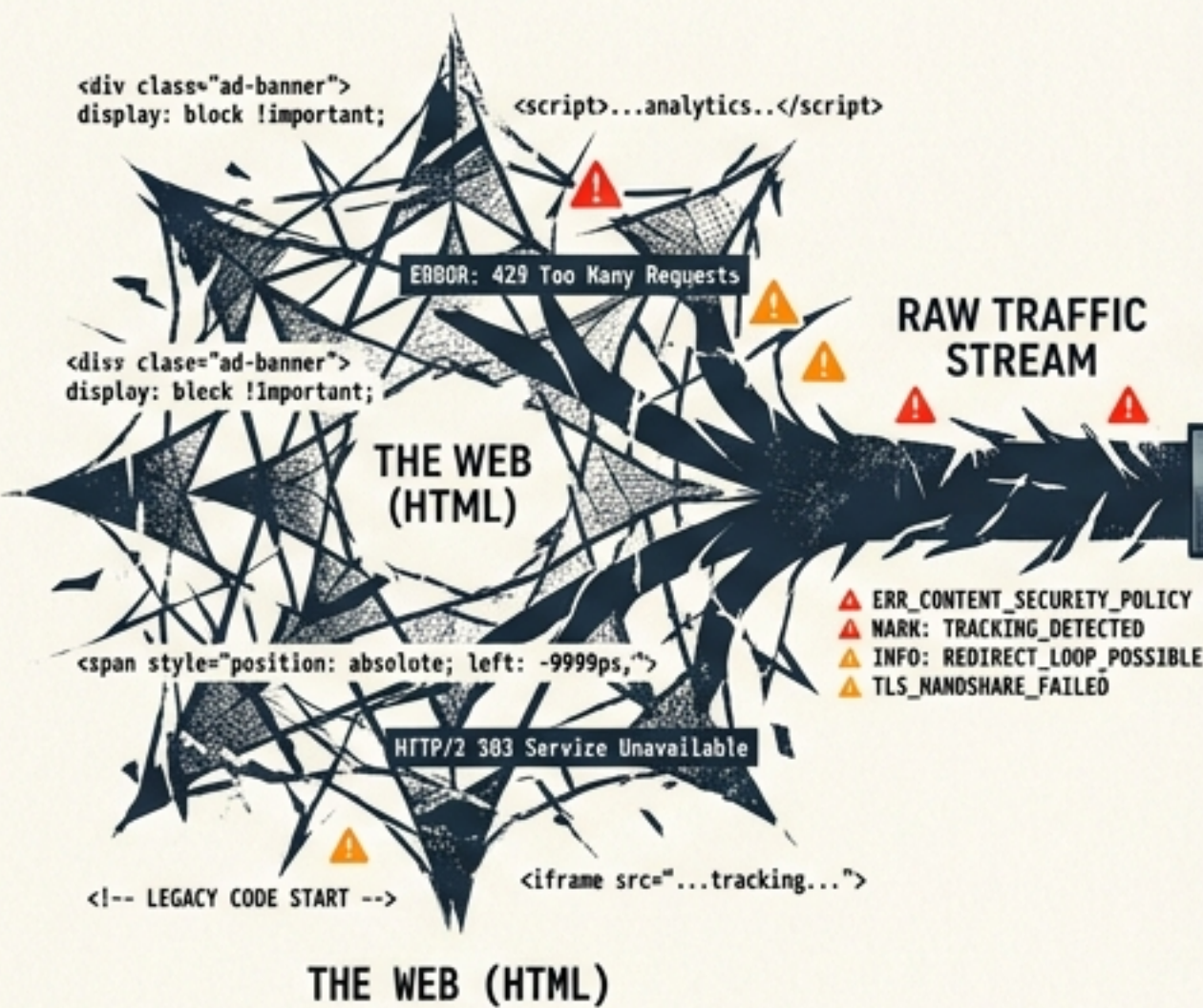


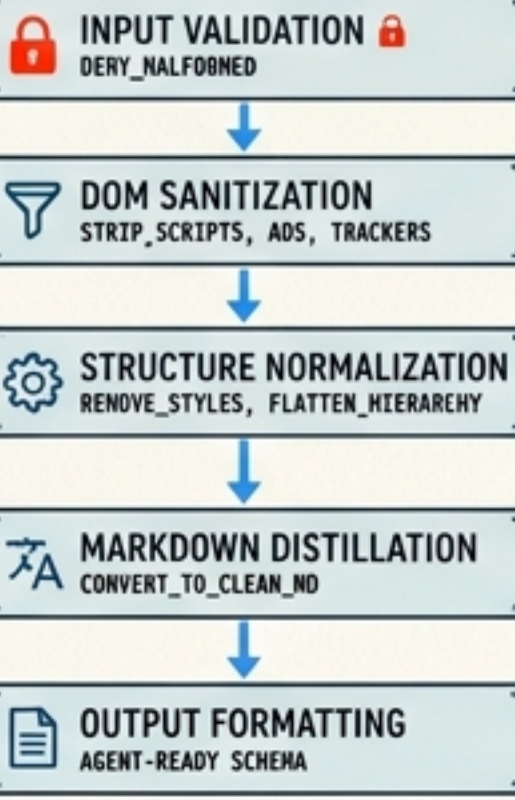
Bulletproof Markdown for Agents

A stateless, self-hosted HTTP proxy that distills raw web pages into clean, agent-ready data.

THE WEB
(HTML)



FILTRATION PROXY



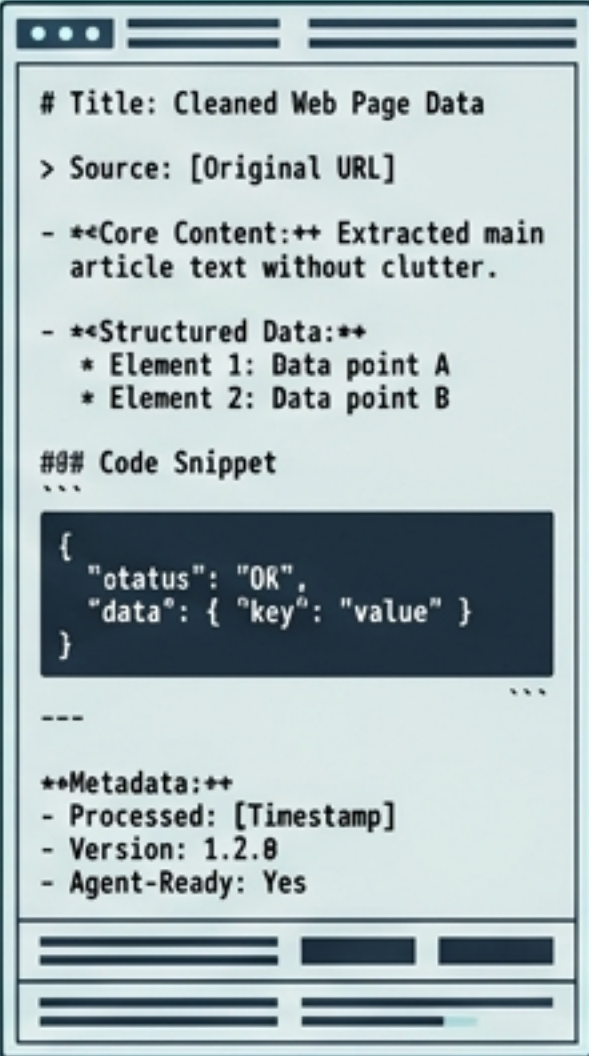
STATUS: ACTIVE (99.9% UPTIME)

SELF-HOSTED (STATELESS) PROXY INSTANCE

DISTILLED
MARKDOWN
STREAM

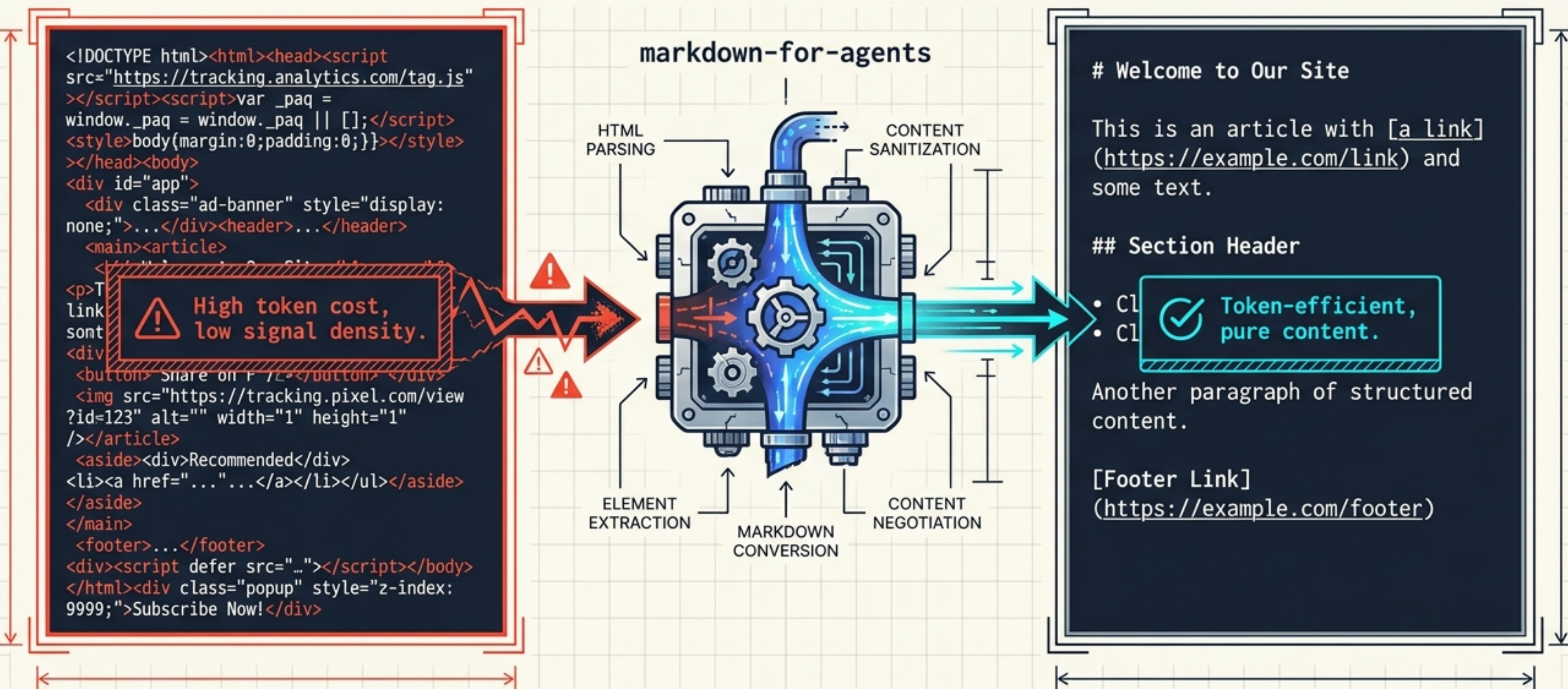
OK: 200 (CLEAN_DATA)
INFO: STRUCTURE_VALIDATED
DATA_FLOW: OPTIMAL

AI AGENT
(MARKDOWN)



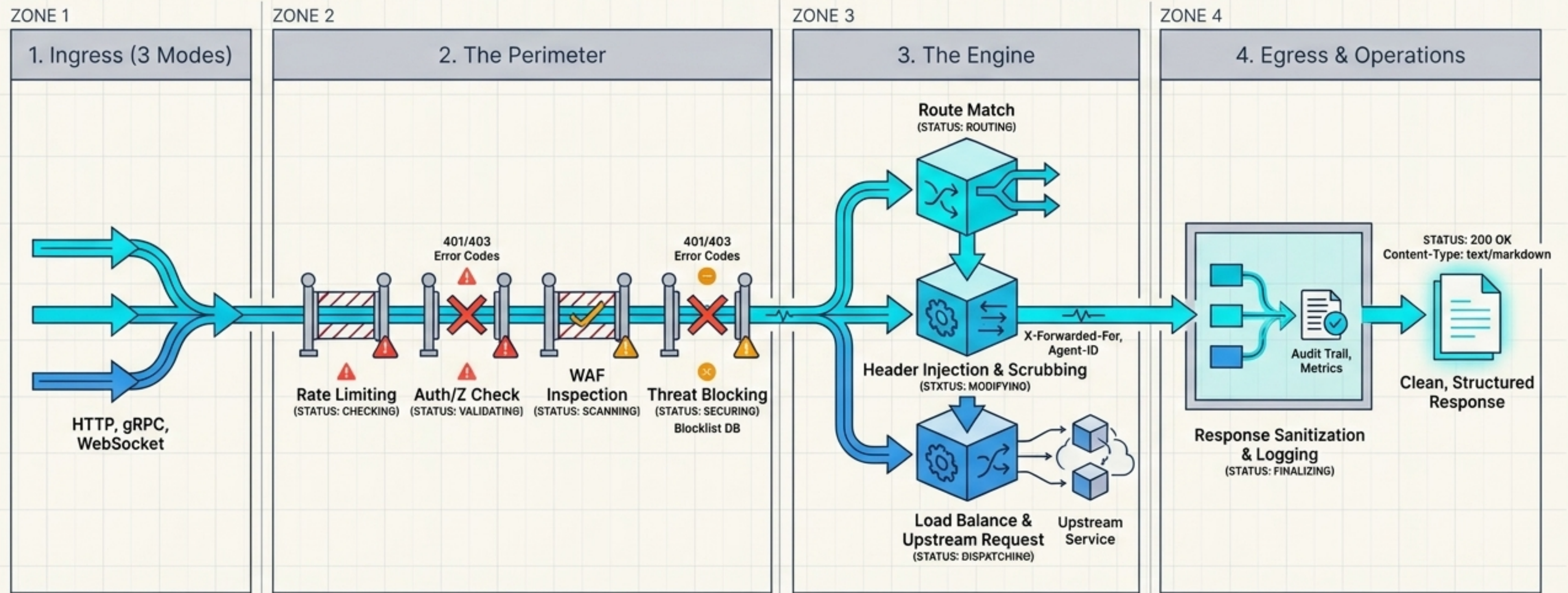
AI agents require structured signal, not HTML noise

The proxy acts as a strict **content-negotiation** layer, fetching arbitrary web pages and returning only clean text/markdown; charset=utf-8 without requiring a database, state, or authentication.



The lifecycle of a proxied request

Every request passes through a rigorous sequence of validation, security checks, and conversion heuristics before returning a single byte to the agent.

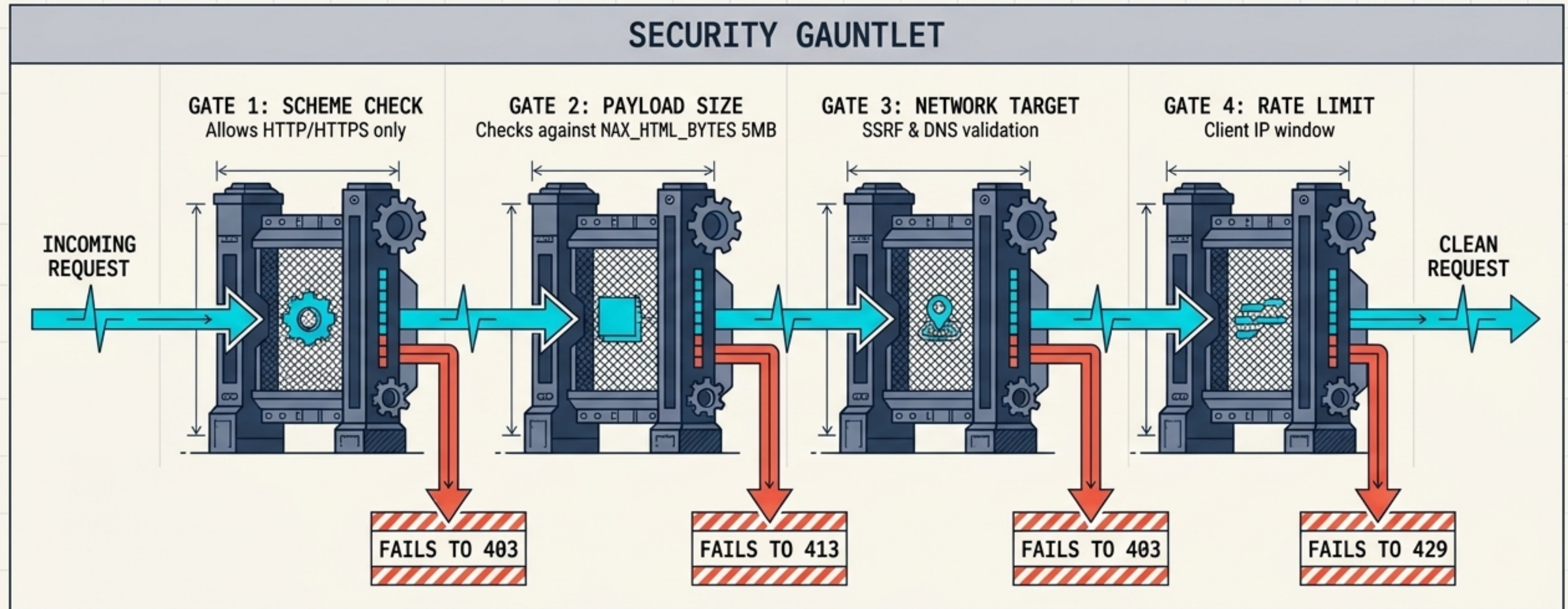


Three discrete ingress modes for variable architectures

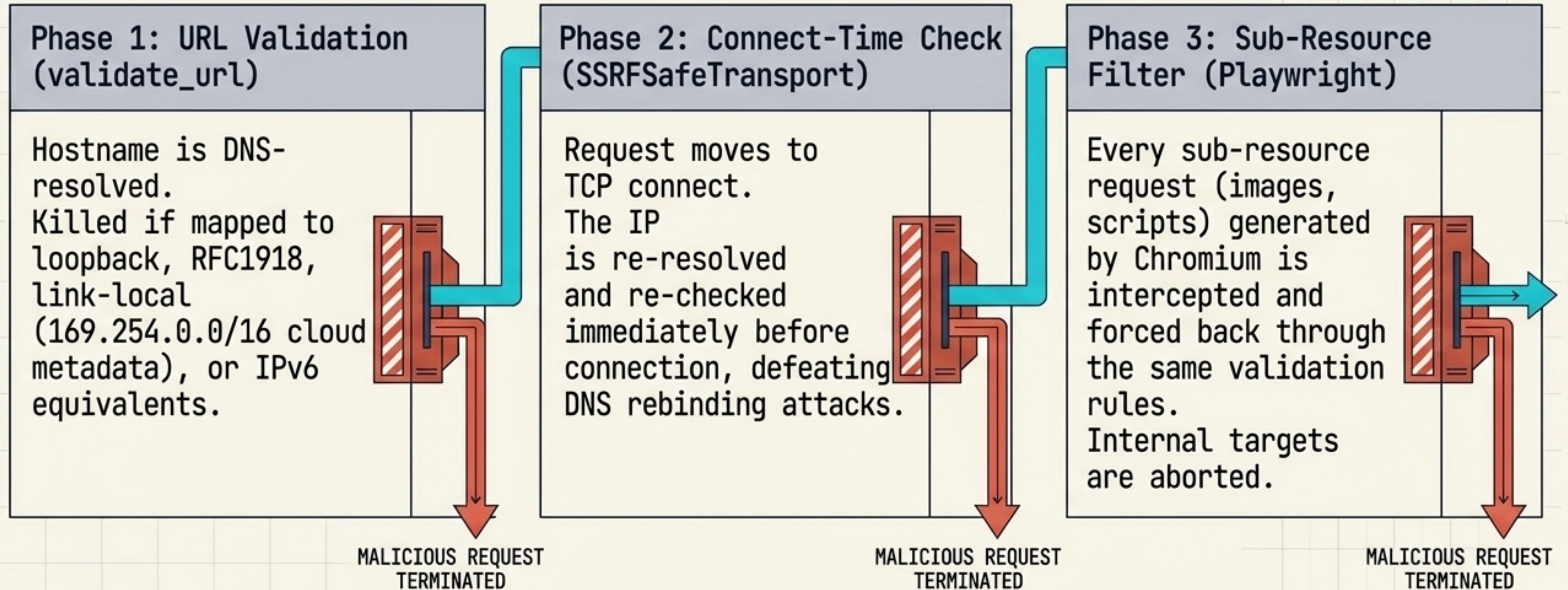
URL-Path Proxy (Mode 1)	Header Proxy (Mode 2)	Sidecar (Mode 3)
Mechanism: Target URL injected into path: GET /https://example.com or ?url= Pipeline: Attempts Native Markdown first, falls back to HTML conversion. Ideal For: Direct agent tool calls.	Mechanism: Request sent to /markdown-convert with X-Original-URL: https://... Pipeline: Always converts HTML (no native attempt). Ideal For: nginx content-negotiation integration (Accept: text/markdown).	Mechanism: ORIGIN_BACKEND env var points to a local app. Pipeline: Always converts HTML. Ideal For: Sitting directly in front of a local backend service.

The Security Perimeter: Pointing at arbitrary URLs requires armor

Because the proxy fetches URLs supplied by autonomous agents, it assumes every request is hostile.

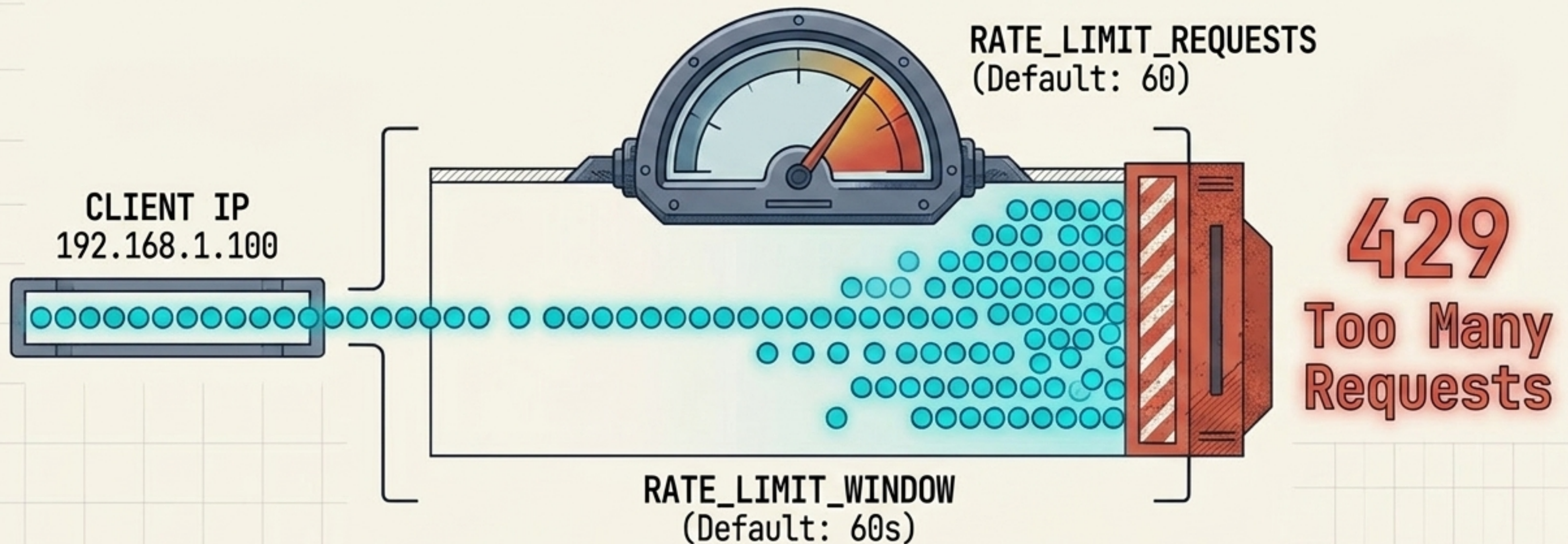


Defeating SSRF and DNS rebinding at the network layer



Per-client rate limiting prevents proxy exhaustion

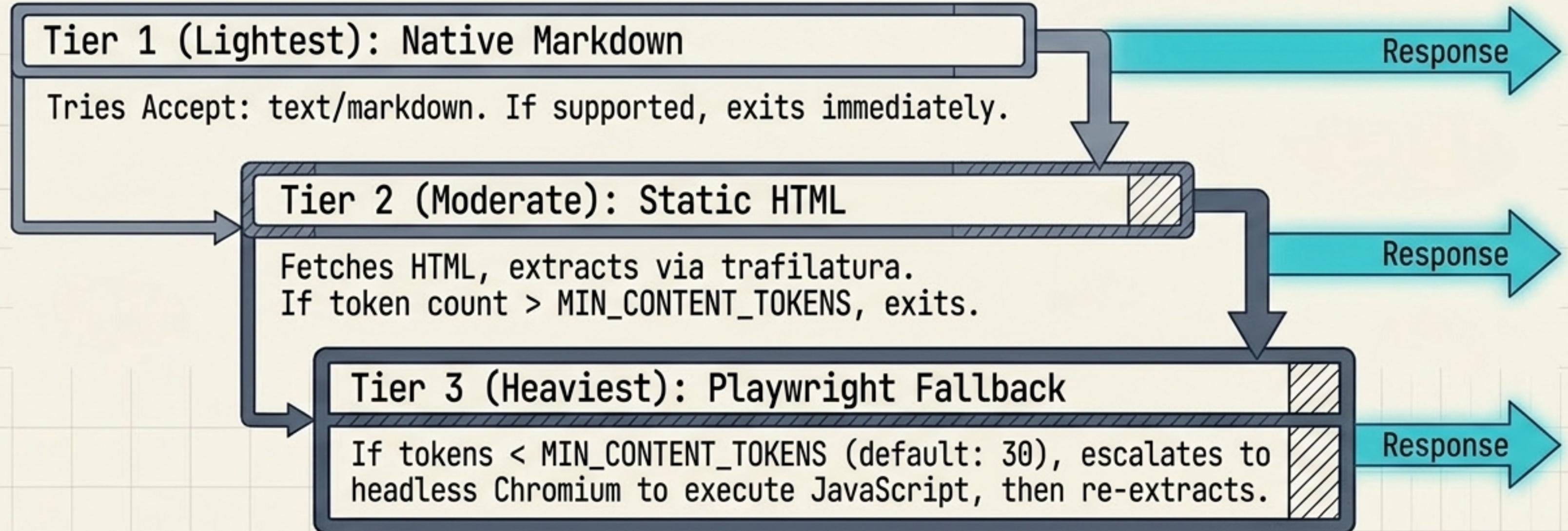
Limiting is enforced per `request.client.host`. If deployed behind a reverse proxy, the proxy must terminate the limit or forward the true client IP, otherwise the service limits the reverse proxy's IP.



The Escalation Pipeline: From cheap headers to heavy rendering

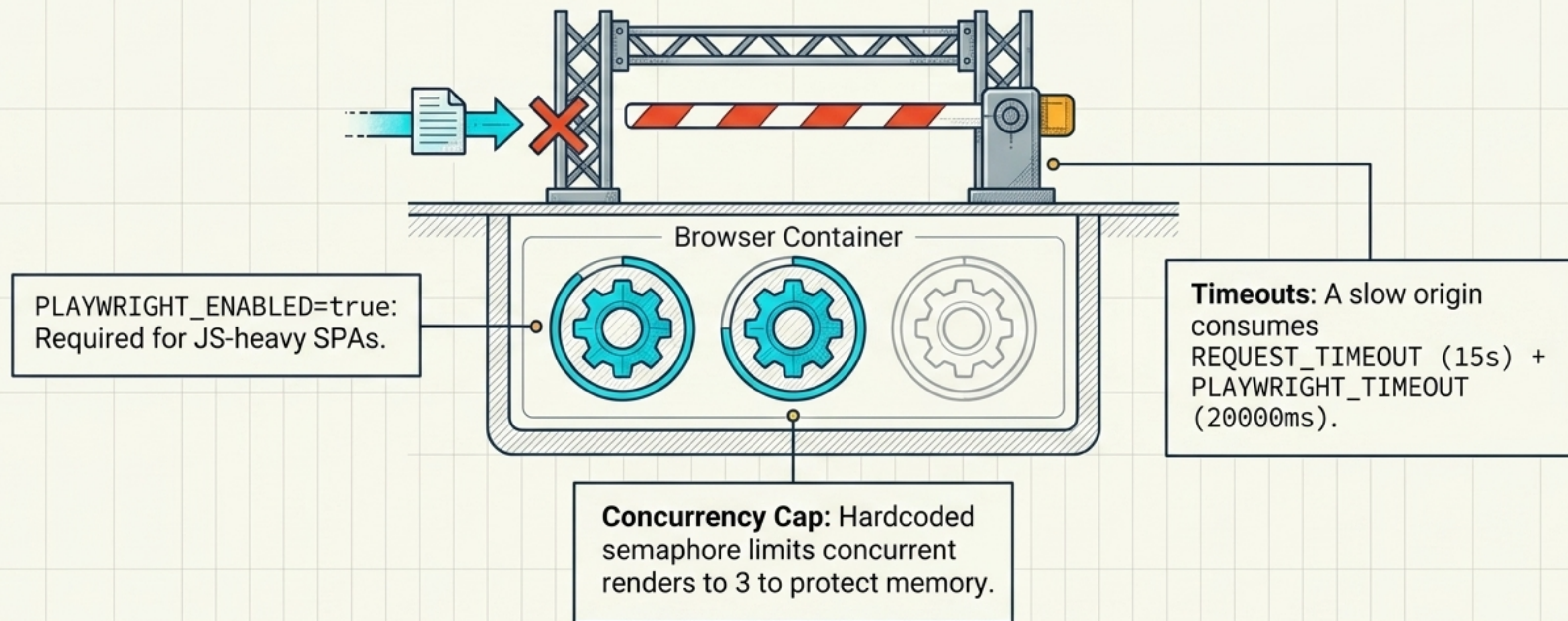
The engine always attempts the most computationally efficient extraction method first, only spending CPU cycles on a full browser render when absolutely necessary.

Conversion Cost Escalation



Bounding the headless Chromium fallback

Rendering is the most expensive path. For workloads strictly converting static blogs or wikis, disabling Playwright saves significant memory.



Egress Response: Clean Markdown with structured metadata

The output is formatted for maximum LLM legibility. HEAD requests return these exact headers with an empty body, allowing agents to perform a cheap token-count or cache probe without downloading the payload.

```
HTTP/1.1 200 OK
```

```
Content-Type: text/markdown; charset=utf-8
```

Headers Block

```
x-markdown-source: converted
```

```
x-markdown-tokens: 450
```

Metadata Headers

```
x-cache: MISS
```

```
---
```

```
title: Sample Page
```

```
description: A brief summary of the content.
```

Injected YAML
Frontmatter

```
---
```

```
# Core Content Begins Here...
```

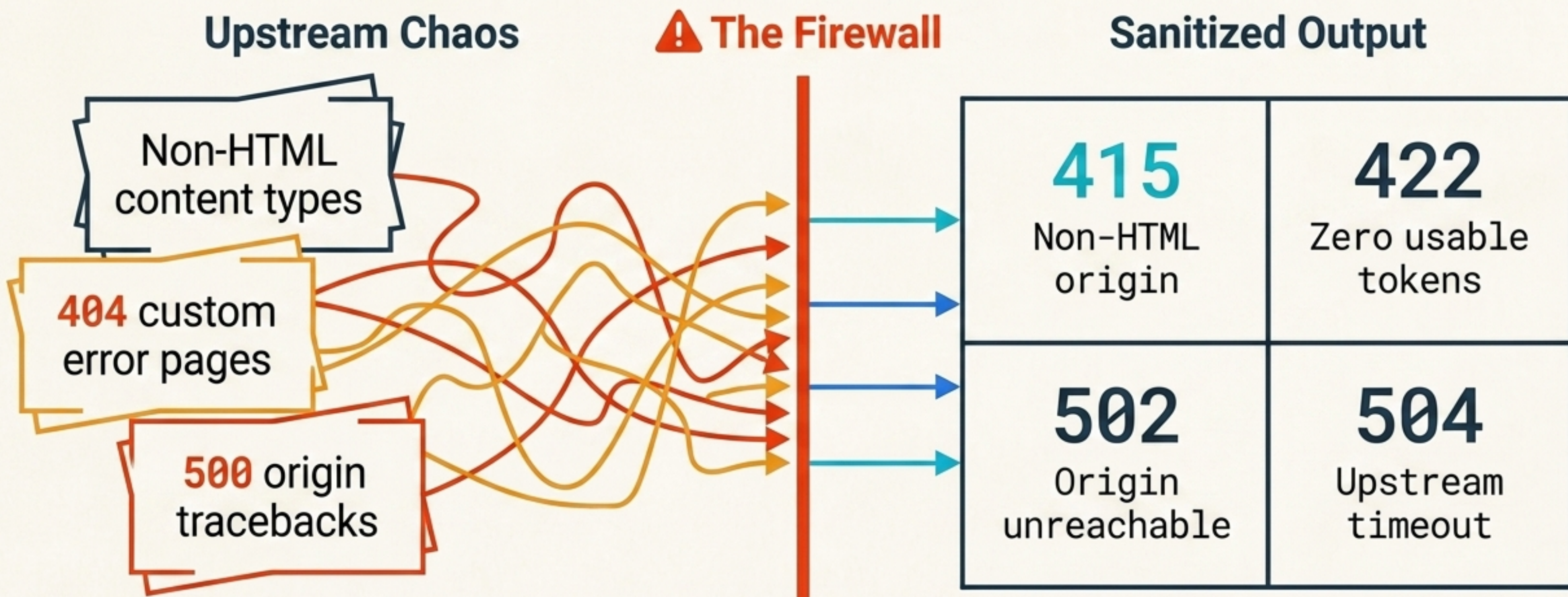

Telemetry injected into every successful response

Metadata allows the consuming application to make programmatic decisions about token budgets and content sourcing without parsing the body.

x-markdown-source	x-markdown-tokens	x-cache	content-signal
Shows how the engine succeeded. Values: native, converted, or rendered.	Estimated token count of the returned document.	Indicates HIT (served from in-process cache) or MISS.	A configurable usage signal string. Default: ai-train=yes, search=yes, ai-input=yes.

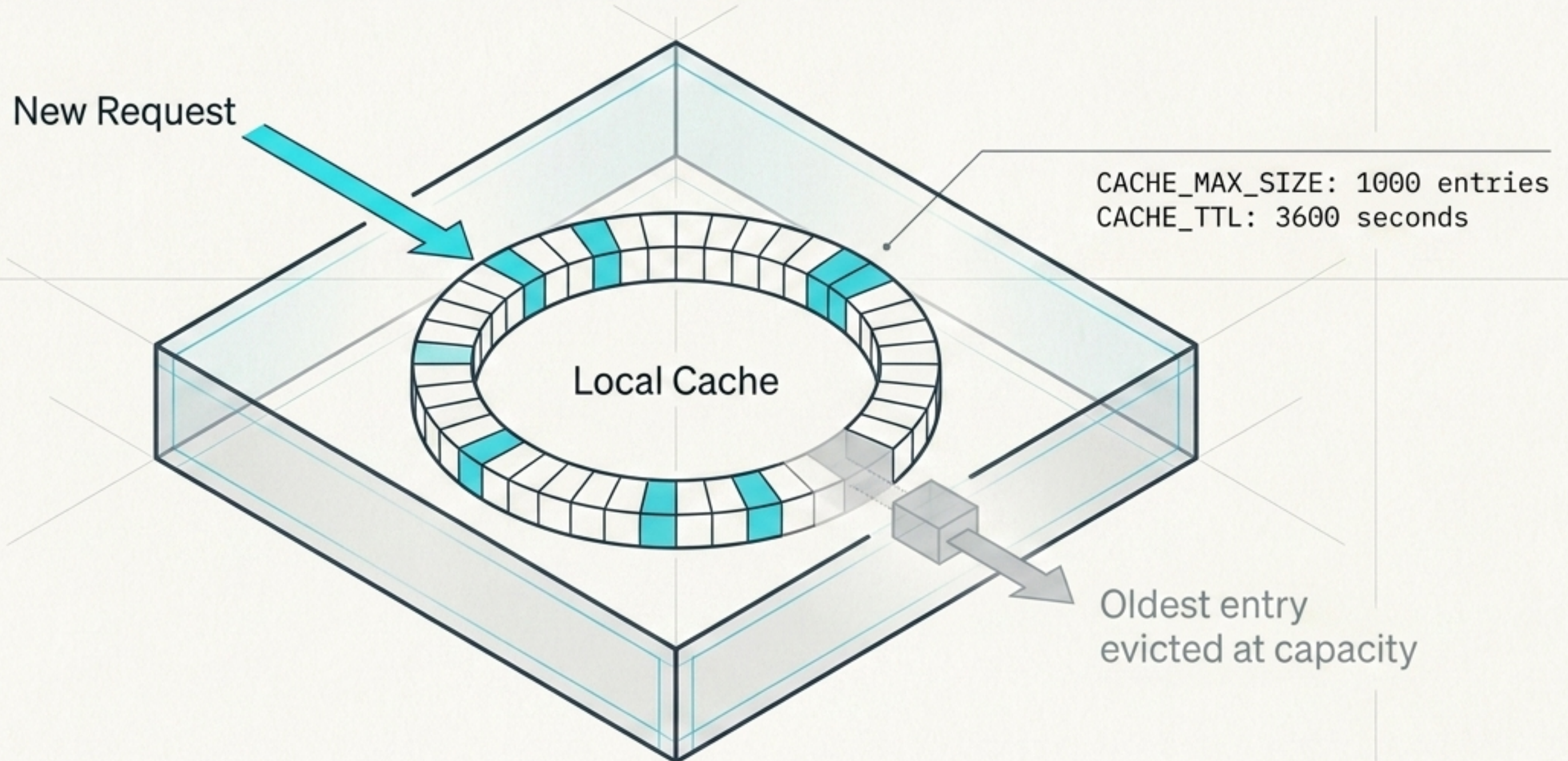
Strict error discipline prevents context window pollution

Non-200 upstream statuses and non-HTML content types are never proxied verbatim. The agent receives predictable, deterministic error states.



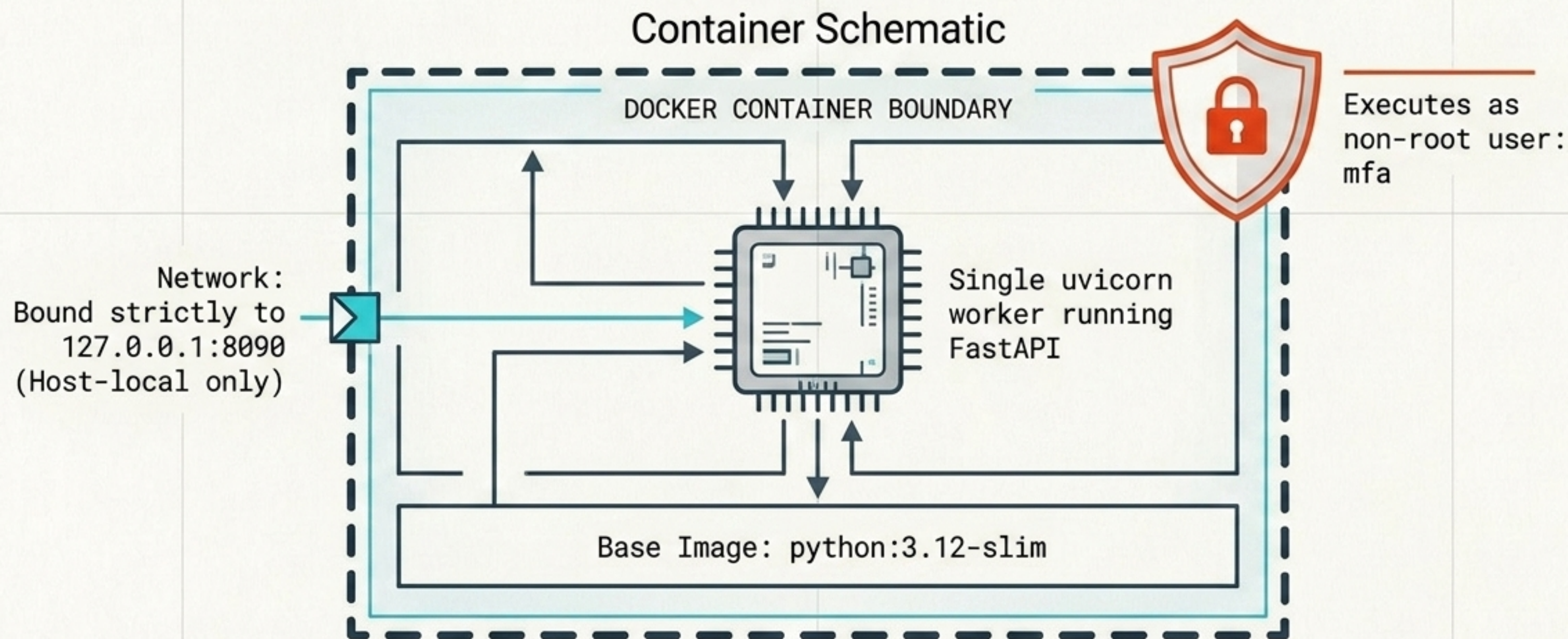
In-process caching accelerates repeat data fetches

The cache is purely in-process per-container and does not share state across replicas. It clears on restart, maintaining the system's strictly **stateless**, zero-database architecture.



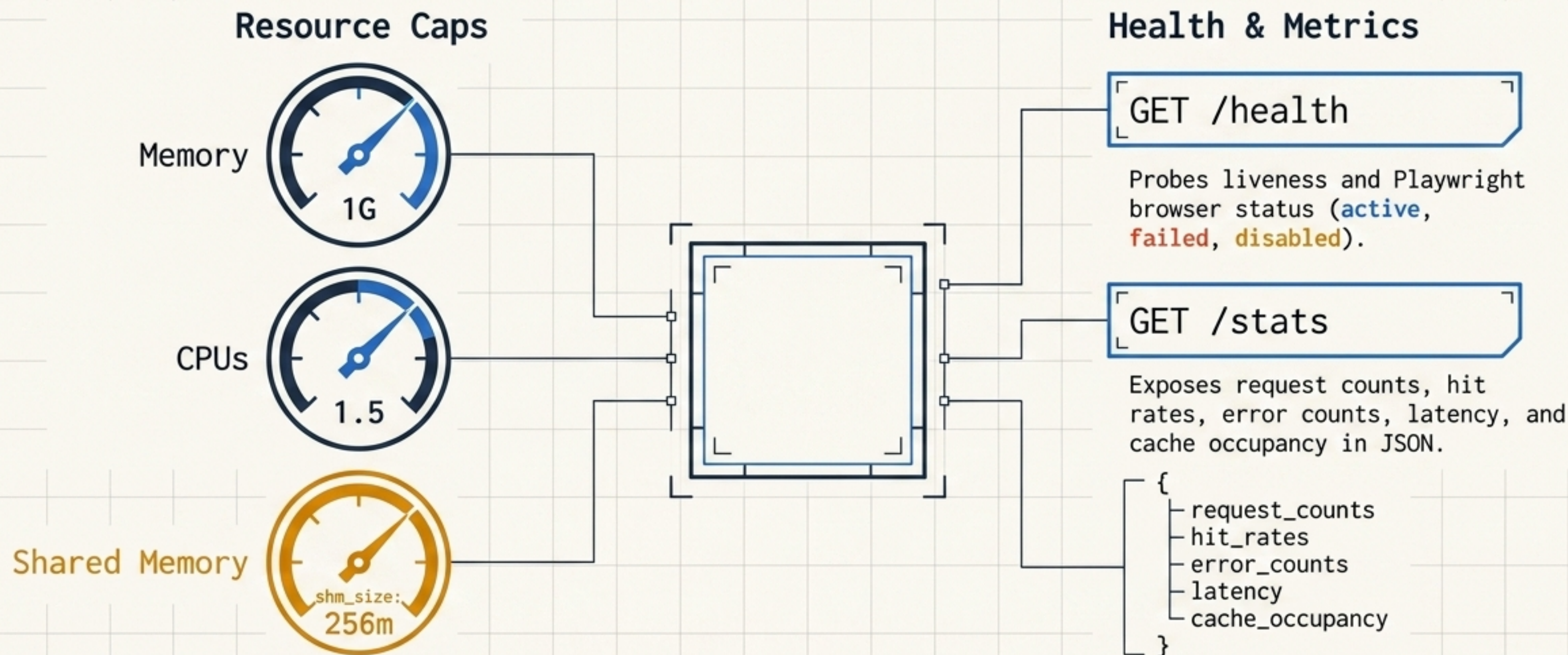
Operational Reality: The Docker containment field

The system is designed to be completely self-contained. No volumes, no database, no external state stores, and no secrets.



Resource limits and container health probing

Production stability relies on strict resource boundaries. The included `docker-compose.yml` pre-configures these safety rails.



A frictionless bridge between the web and the agent

By isolating the complexities of web scraping, headless rendering, and network security into a single, stateless Docker container, **bulletproof-markdown-for-agents** allows AI platforms to consume the internet safely, cheaply, and reliably.

