

bulletproof-design — System Overview

Architectural breakdown, data modeling, and security protocols for the self-hosted design library.

Unified Container Orchestration

Control Plane

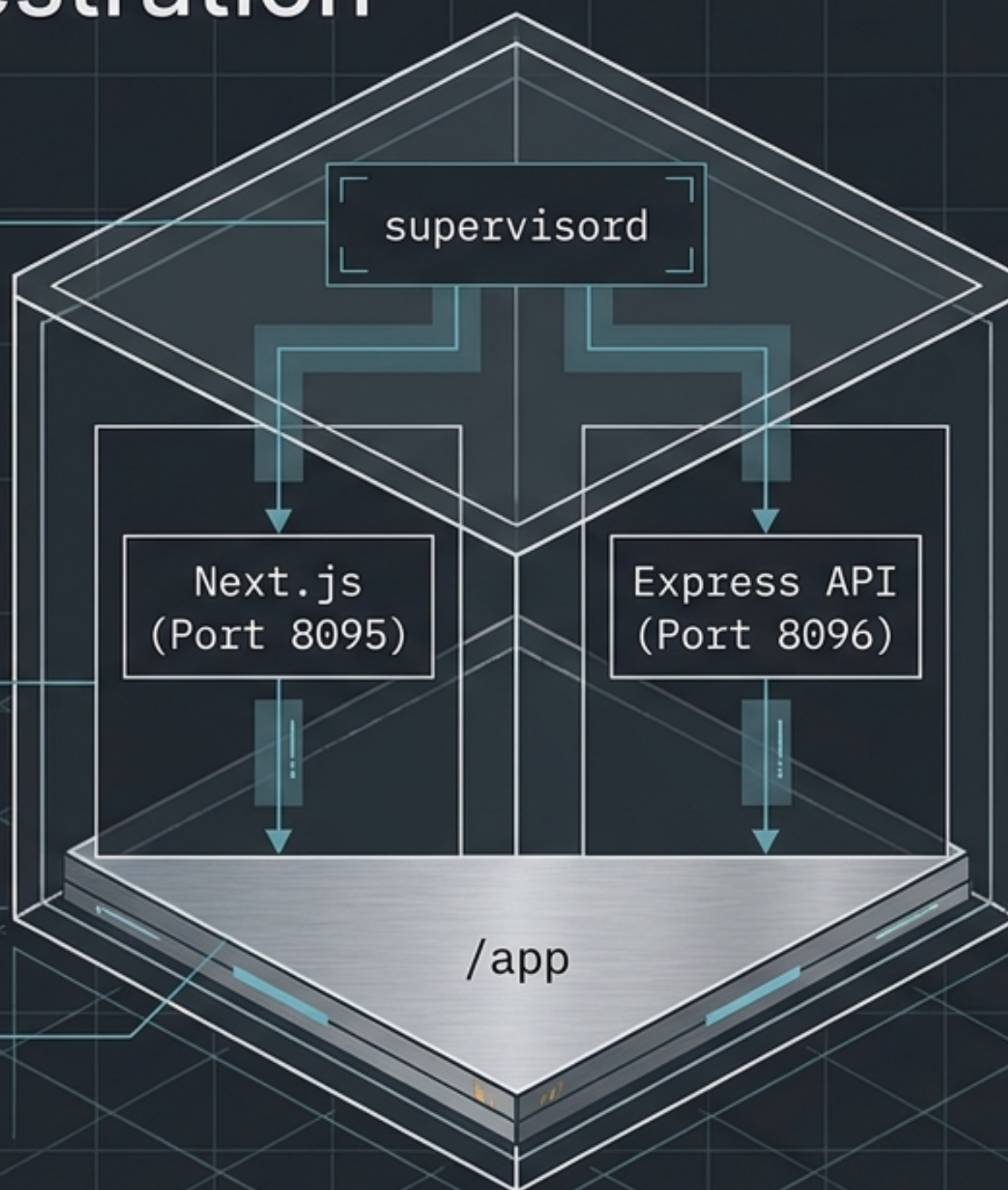
supervisord triggers both services via a single docker compose up --build command.

Execution Lanes

Two distinct services run in parallel within the isolated container boundary.

Shared Environment

Both servers operate strictly within the /app working directory.



The Dual-Server Architecture Split

Next.js Application (Port 8095)

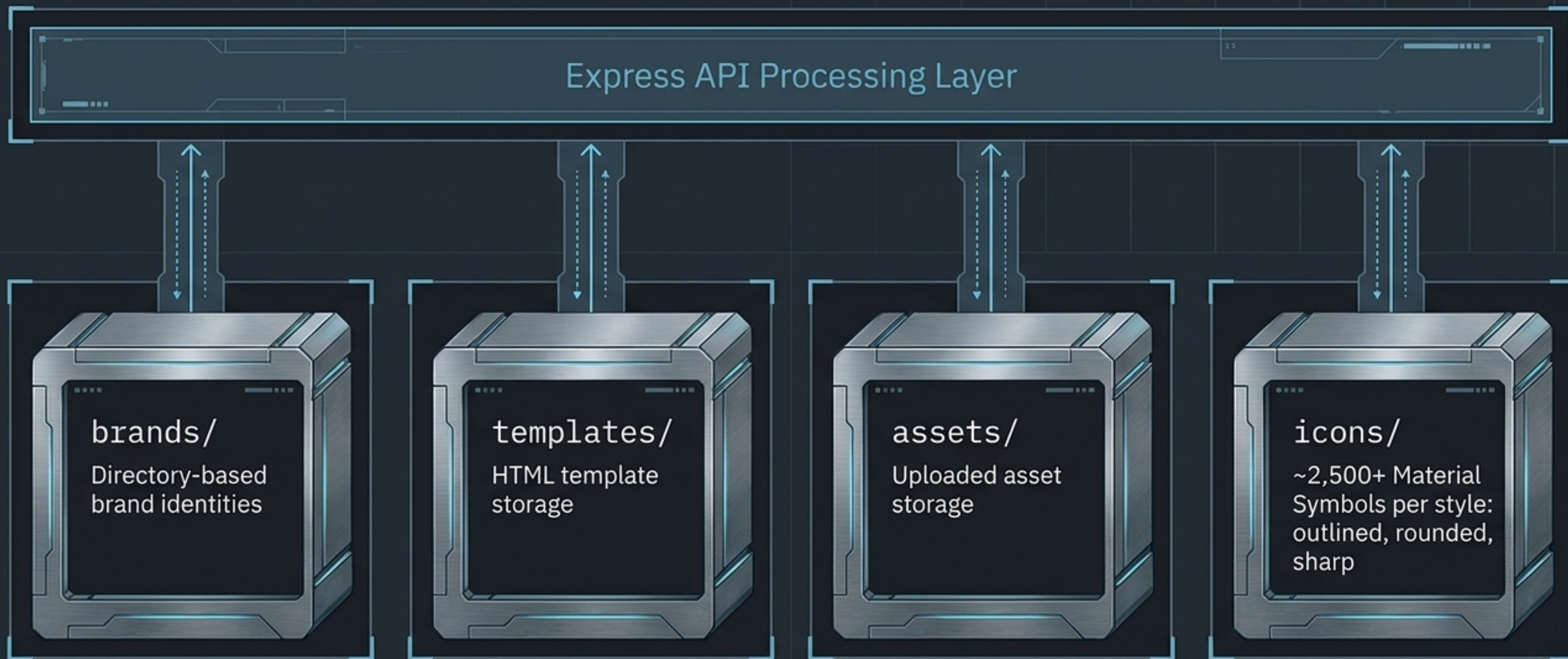
Responsibility	SSR for public portal & admin UI, App Router client navigation.
Architecture	Hydration-safe (no direct window/document access outside useEffect).
Routing	Route groups (/portal, /(admin)).

Express API (Port 8096)

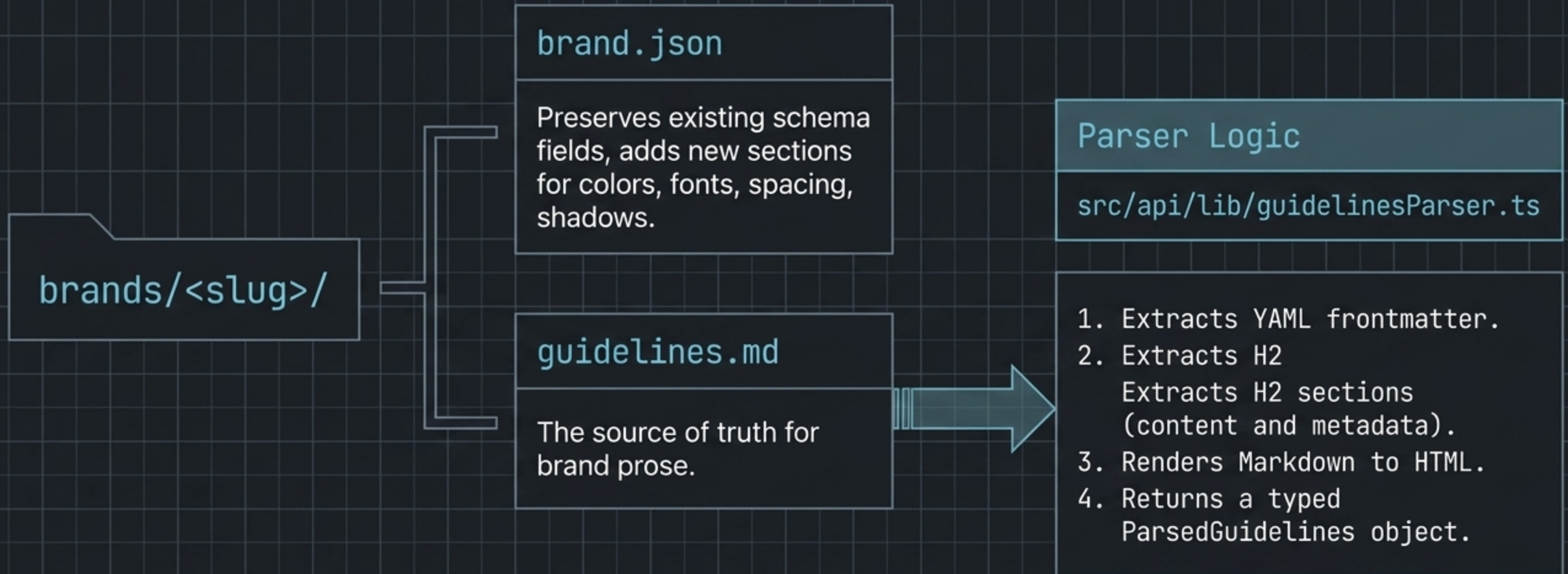
Responsibility	File-system backed REST API, data management.
Architecture	In-memory indexes (rebuilt from disk at startup).
Security	Helmet CSP headers, rate limiting, optional API key auth.

File-System Backed Volume Mounts

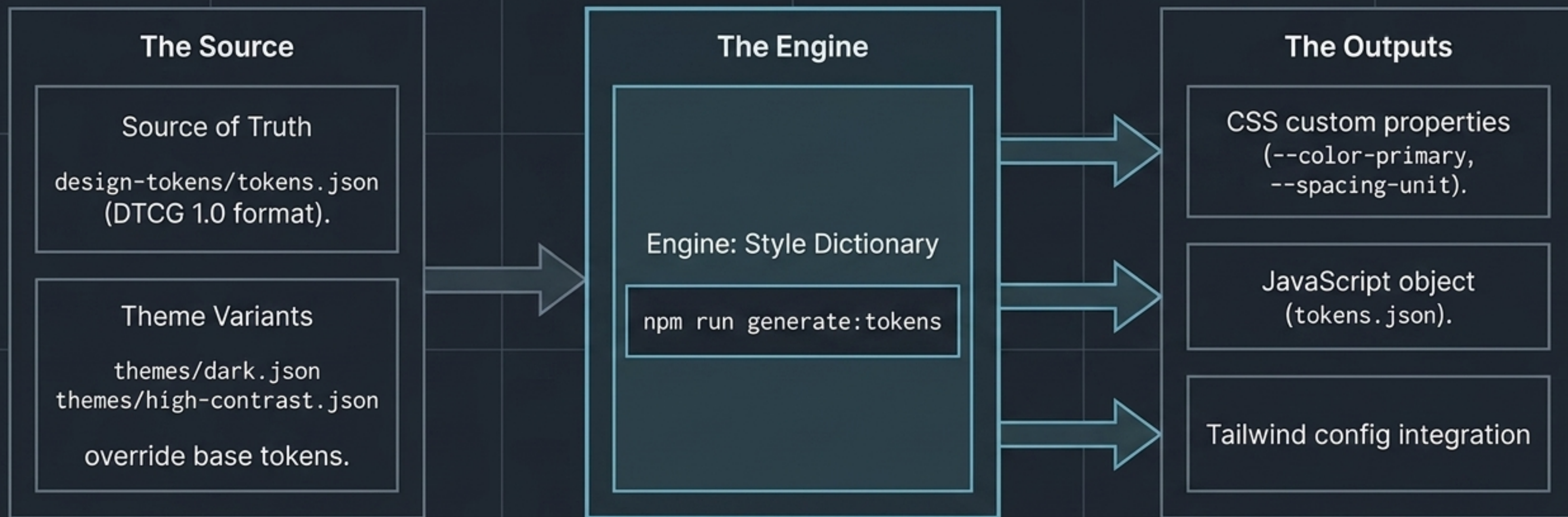
The system eschews a traditional database in favor of direct file-system backing via Docker mounted volumes.



Anatomy of the Brand Directory

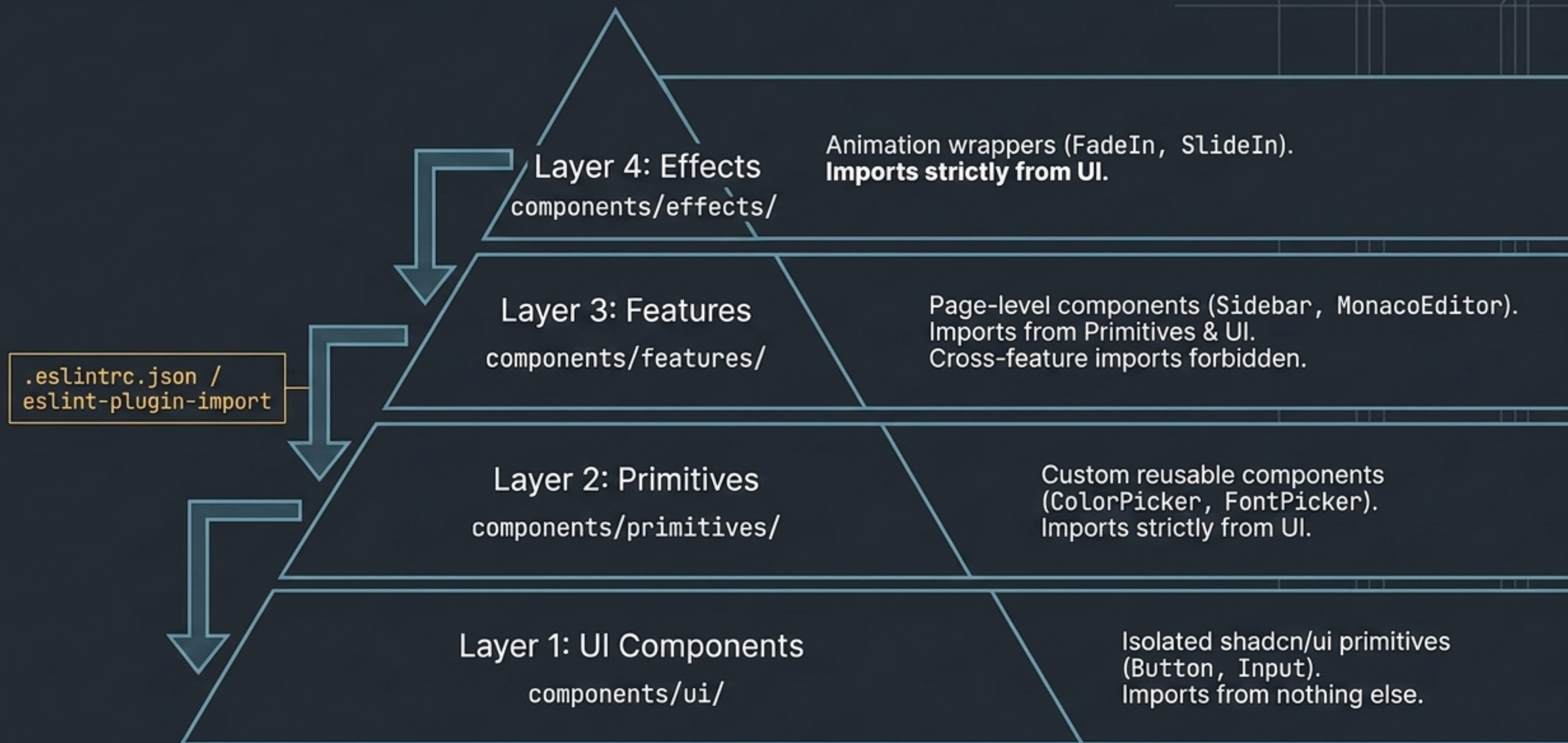


Token Compilation Pipeline



Enforcement: ESLint warns against hardcoding hex colors (e.g., #1f2937) in TSX, enforcing CSS variable usage.

Four-Tier Enforced Component Hierarchy



Defense in Depth Security Posture

Outer Ring: Supply Chain Guardrails

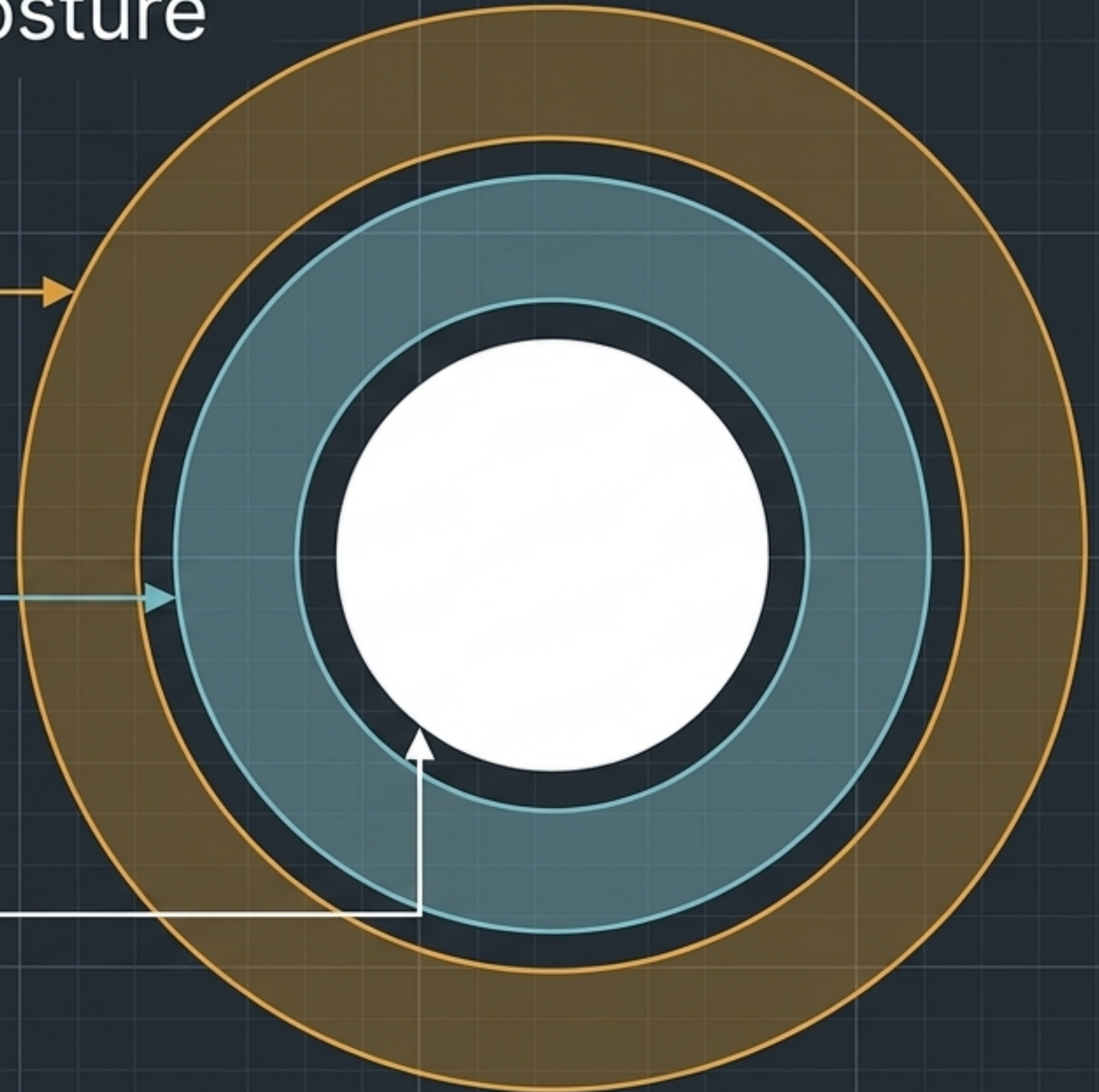
- SBOM Generation.
- `audit:check gate` runs inside the Docker image build to block HIGH/CRITICAL advisories on production dependencies.

Middle Ring: Network & Access

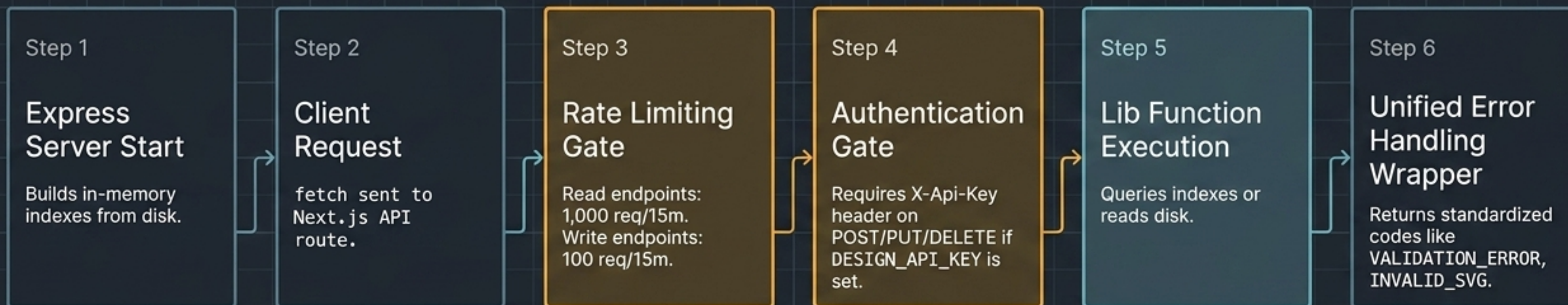
- Rate limiting middleware.
- CORS origin restrictions (`CORS_ORIGIN` env var).

Inner Core: Content & Execution Integrity

- Strict Content Security Policy (CSP).
- Aggressive SVG Sanitization.
- Constant-time comparison (`safeCompare`) for API key auth to prevent timing attacks.



API Request Lifecycle & Throttling



Content Sanitization & Execution Policies

SVG Sanitization (via sanitize-html 2.17.3+)

- Applied to logo uploads (Max 1MB) and GitHub syncs (npm run icons:sync).
- **Action:** Strips all `<script>` tags, removes event handlers (on*), blocks external entity references (XXE prevention).
- **Allows:** Standard SVG elements, attributes, and MathML only.



Content Security Policy (CSP)

Production

```
script-src 'self' 'nonce-{random}'  
'strict-dynamic'
```

(No **unsafe-inline**). Nonce regenerated per request.

Development

Retains `unsafe-inline` and `unsafe-eval` solely for HMR and Monaco editor functionality.

Zero-Downtime Rollout Levers

Request-Scoped Flags (Immediate Effect)

A DISABLE_PORTAL

Returns 404 on portal pages; blocks portal API reads. Admin remains functional.

B PORTAL_INDEX

Defaults unset (secure noindex,nofollow). Set to **1** to enable public search indexing on prod.

C DESIGN_API_KEY

Requires API key **auth** for **write** endpoints.

Startup-Scoped Flags (Requires Restart)

A DISABLE_ICONS

Unmounts icon routes; shows empty state.

B DISABLE_ENRICHED_COMPONENTS

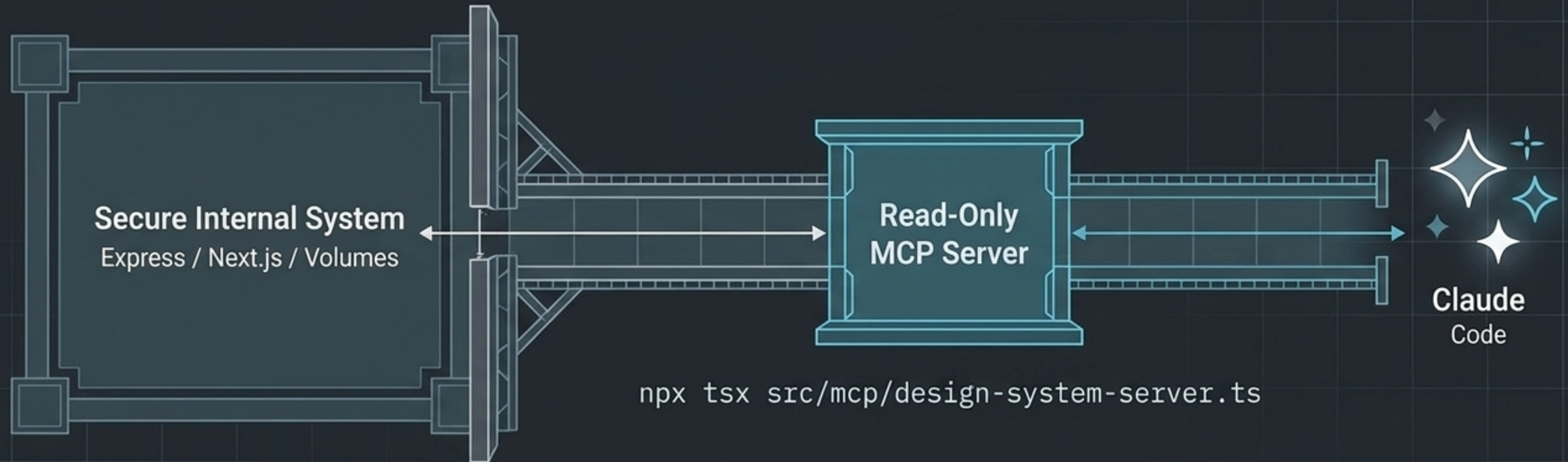
Returns legacy 14-item component manifest.

Script-Scoped Flags

A GITHUB_TOKEN

Raises GitHub API limits for Material Symbols sync.

AI Extensibility via Model Context Protocol (MCP)



Purpose

An isolated, read-only MCP server running on stdio that exposes design system data to AI tools.

Mechanics

Accepts inputs via Zod schemas, respects feature flag checks (`DISABLE_*`), and returns structured JSON error responses.

The 10 AI-Accessible Design System Tools

Component & Asset Tools - Legacy

`get_component /
search_components`
Lookup full registry
entries.

`get_asset`
Fetch asset metadata.

`get_tokens`
Retrieve categorized
design tokens.

`get_composition_rules`
Fetch component
layout constraints.

`get_components`
Fetch asset size
shadows.

Brand & Identity Tools - New Expansion

`get_brand_identity`
Colors, fonts, spacing,
shadows.

`get_brand_guidelines`
Parsed guidelines:
prose and sections.

`get_component_spec`
Full spec with props,
variants, examples.

`search_icons`
Query ~7,500 Material
Symbols SVGs.

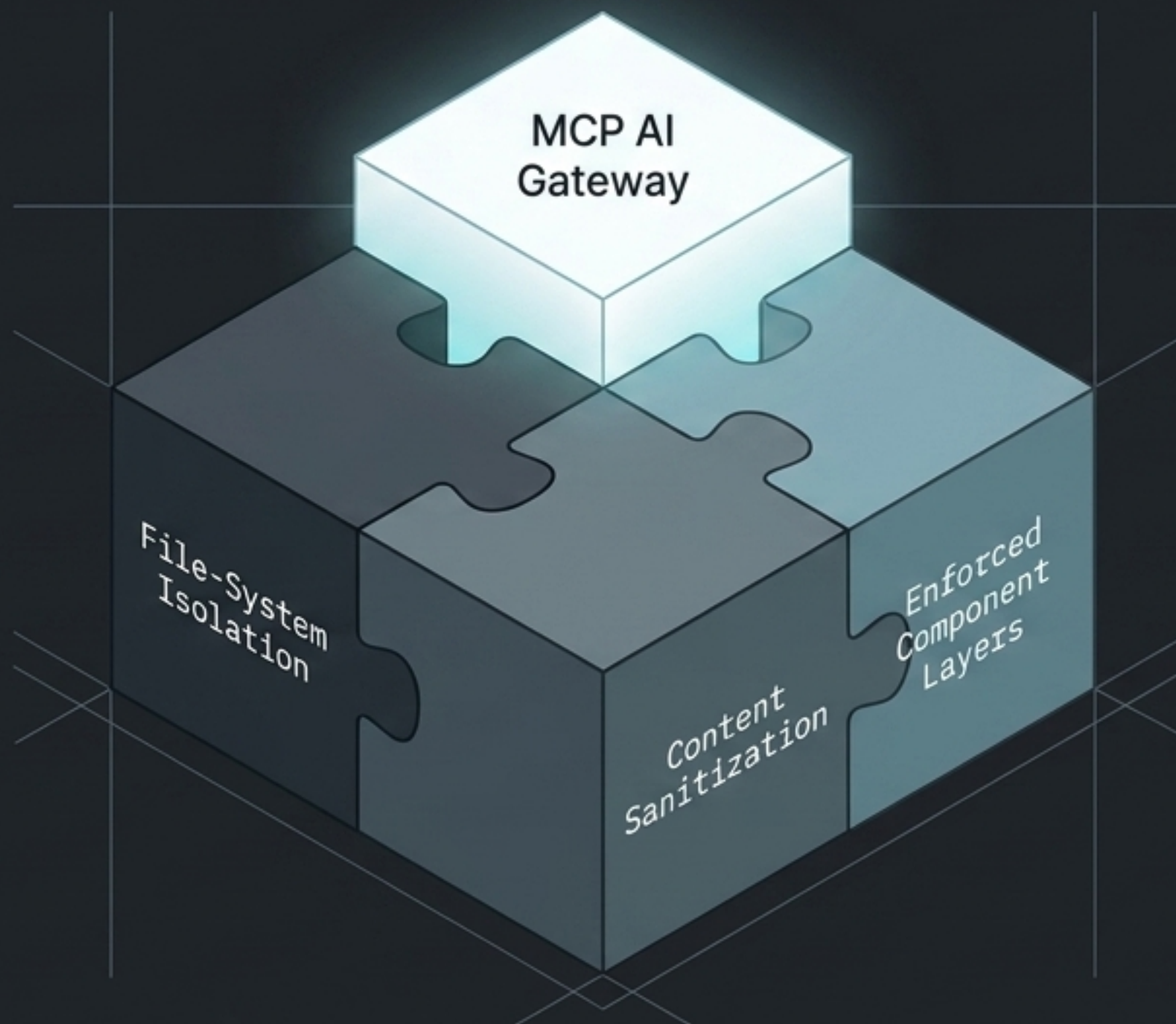
`get_icon`
Fetch specific style
SVGs.

Synthesis: The Strict Base Enables the Flexible Future

The architecture's aggressive 'bulletproof' choices make AI extensibility inherently safe.

Because the data model relies on isolated file-system backing rather than an exposed database...

Because payloads are restricted by strict **Content Security Policies**, constant-time compare auth, and relentless SVG sanitization...



...The MCP Server can safely expose the entire design system repository to autonomous tools like Claude Code without **risking system integrity** or supply chain poisoning.

Precise engineering yields fearless integration.