INDEPENDENT TECHNICAL DUE DILIGENCE

Open-Generative-AI & DASCA

Two unrelated targets assessed against a technical due-diligence lens: code quality, architecture, real versus claimed capability, supply-chain exposure, security posture and project health. One is a 25k-star open-source repository. The other is a commercial WebGL creative tool. They could not have scored more differently.

REPORT DATE	TARGETS	FINDINGS	CRITICAL / HIGH
29 July 2026	2	24	3 / 7
TESTING MODE			
Passive only			

Executive summary

The two targets are **unrelated**. DASCA is not a fork, rebrand or deployment of Open-Generative-AI; they share no code, no organisation and no infrastructure beyond both sitting behind Cloudflare. Each is assessed on its own merits.

HEADLINE **Open-Generative-AI is a paid API client marketed as self-hosted software**

The repository's GitHub description reads "*Free AI image & video generation studio with 500+ models… **Self-hosted**, MIT licensed.*" In practice the application cannot generate anything until the user supplies a **MuAPI** key and pays MuAPI per generation. Every server route proxies to `api.muapi.ai`, and the entire codebase declares exactly one environment variable: `MUAPI_KEY`.

MuAPI is not a neutral third party. Its MCP server is published under **SamurAIGPT** — the repository author's own GitHub organisation — carrying an explicit marketing attribution campaign (`utm_campaign=muapi-mcp-server`). The repository functions as a customer-acquisition funnel for a commercial service owned by the same party, and this relationship is disclosed nowhere in the README or licence.

HEADLINE **The 25,052 stars did not accrue to this project**

GitHub records the repository object as created **9 May 2023**. The first commit in its git history is dated **10 February 2026** — a 33-month gap — and reads "*Initial commit: Scaffold Open-Higgsfield-ai with Vite and README.*" The project has been rebranded at least once (Open-Higgsfield-AI → Open-Generative-AI) while retaining the container's accumulated social proof. Advertised model counts escalated **200+ → 420+ → 500+** across five months of the same codebase.

HEADLINE **DASCA is a well-built product with fixable operational gaps**

DASCA presents genuine engineering depth: WebGL2 shader chains, a node-based signal-routing graph with audio band analysis, a live GLSL playground and canvas recording. Its security headers are among the stronger configurations observed in independent web apps — a restrictive CSP with no `unsafe-inline` or `unsafe-eval` in `script-src`, TLS 1.2/1.3 only, and no exposed sourcemaps or secrets.

The defects are configuration-level, not architectural. Most notably, **DASCA's own CSP blocks its own Google Analytics** — verified in a real browser — meaning property `G-2VJGNYXSNB` has been collecting nothing.

Scorecard

OPEN-GENERATIVE-AI

PROVENANCE & TRANSPARENCY	GOVERNANCE / BUS FACTOR	SECURITY POSTURE	CODE QUALITY
2/10	2/10	3/10	3/10

CAPABILITY ACCURACY	MAINTENANCE ACTIVITY	LICENSING HYGIENE	OVERALL
4/10	5/10	7/10	3.4/10

DASCA

PRODUCT CRAFT	TRANSPARENCY	SECURITY POSTURE	OPERATIONAL MATURITY
9/10	8/10	7/10	6/10

PERFORMANCE	PRIVACY / COMPLIANCE	OVERALL
6/10	5/10	7.0/10

Scope & method

Authorisation boundary. DASCA was assessed **passively only**. All observations derive from publicly served content: response headers, TLS negotiation, DNS records, published JavaScript bundles, and a single ordinary page load in an instrumented browser. No authentication testing, input fuzzing, injection probing, endpoint enumeration against the server, or load generation was performed. Where a finding would require active testing to confirm, it is marked INDICATED rather than asserted.

ACTIVITY	TARGET A — REPO	TARGET B — DASCA
Full-history git clone & log analysis	Yes — 330 commits	N/A — closed source

ACTIVITY	TARGET A — REPO	TARGET B — DASCA
Secret scan across all history blobs	Yes	Bundles only
Dependency audit (`npm audit`)	Yes — 1,168 packages	N/A
Static source review	Yes	Minified bundles only
HTTP headers & CSP analysis	From source	Live
TLS negotiation test	N/A	Yes — TLS 1.0–1.3
DNS / email authentication records	N/A	Yes
Instrumented browser load	N/A	Yes — console + network
Active security probing	N/A	**Not performed**

Target A — Open-Generative-AI

REPOSITORY	STARS / FORKS	LICENCE
Anil-matcha/Open-Generative-AI	**25,052 / 4,381**	**MIT**

LANGUAGE	COMMITS	SOURCE FILES
JavaScript	**330**	**148 (94 JS/JSX)**

CLONE SIZE	LAST PUSH
125 MB	**28 Jul 2026**

The project presents itself as a self-hosted, MIT-licensed studio for AI image and video generation with access to 500+ models, positioned as a free alternative to commercial platforms. It ships a Next.js web application and Electron desktop builds for macOS, Windows and Linux.

Provenance & rebranding

HIGH **Repository object predates its code by 33 months**

The GitHub API reports the repository was created `2023-05-09`. The earliest commit reachable in the full history is `2026-02-10`. A repository that has been renamed retains its stars, watchers, forks and creation date; its git history does not. The 25,052 stars therefore cannot be attributed to the code currently in the repository, and star count should be discounted entirely as a quality or adoption signal here.

MEDIUM **Serial rebranding tracks trending products**

The initial commit scaffolds "*Open-Higgsfield-ai*". The `master` branch still carries the original README — "*Open Higgsfield AI — Open-Source Alternative to Higgsfield AI*" — advertising 200+ models and linking release binaries under the old `Open-Higgsfield-AI` repository name. The current `main` branch advertises 500+ models under a generic name. The `package.json` description still reads "*Open-source alternative to HF AI*", a leftover from the Higgsfield era.

Advertised model counts across the same codebase: **200+** (Feb) → **420+** → **500+** (Jul).

Claims vs. reality

CLAIMED	FOUND
"Self-hosted"	Generation requires a paid `api.muapi.ai` key. All eight API routes proxy to `MUAPI_BASE = 'https://api.muapi.ai'`; `middleware.js` rewrites `/api/v1/*` to the same host. The only environment variable in the entire codebase is `process.env.MUAPI_KEY`.
"Free"	The software is free; the service it depends on is billed per generation. Users hit a "*Muapi API Key Required*" modal before first use.
"500+ models"	**418** unique model identifiers across all registries (`models_dump.json`, `packages/studio/src/models.js`, `src/components/ImageStudio.js`). `models_dump.json` itself contains 51 text-to-image entries. The count is inflated by roughly 20%.
"Midjourney, Sora, Veo"	Identifiers such as `midjourney-v7-text-to-image` exist in the registry. Midjourney publishes no public API; access is brokered through MuAPI as an intermediary. Adopters inherit the terms-of-service risk of that arrangement.

Where the claim holds partially. Local inference is not entirely absent. The codebase contains `src/lib/localInferenceClient.js` and `src/components/LocalModelManager.js`, supporting `sd.cpp` (sd1 / sdxl / z-image) and Wan2GP models, and the four existing tests all target this path. However `build/local-ai/` ships containing only a README, and local inference covers a small fraction of the advertised model catalogue. "Self-hosted" is defensible for a handful of models and misleading for the headline figure.

Undisclosed commercial conflict

CRITICAL **The project funnels users to a service the author controls**

MuAPI is presented throughout the codebase as a neutral upstream provider. Evidence indicates it is connected to the repository author:

- `muapi-mcp-server` is published at `github.com/SamurAIGPT/muapi-mcp-server`. SamurAIGPT is an organisation associated with the repository author, Anil Matcha.
- That repository's homepage field is `https://muapi.ai?utm_source=github&utm_medium=about&utm_campaign=muapi-mcp-server` — a deliberate marketing-attribution campaign, not an informational link.
- All three git submodules point to repositories under the author's own accounts: `SamurAIGPT/Vibe-Workflow`, `Anil-matcha/Open-Poe-AI`, `Anil-matcha/Open-AI-Design-Agent`.
- `vadoo.tv` — another property associated with the author — is referenced in application source.

Neither the README (638 lines, 40 KB) nor the LICENCE discloses any commercial relationship. A user reasonably reading "free, self-hosted, open-source alternative" is not informed that the recommended and only functional backend is a paid service connected to the same author. For due diligence purposes this is the single most material finding on this target: the open-source project is a distribution channel for a commercial API.

Security

CRITICAL **API key in localStorage behind a self-defeating CSP**

The MuAPI key is written to `localStorage` under key `muapi_key` and read back at 15 call sites across the studio components. Simultaneously, the application's own `middleware.js` sets:

```
// middleware.js — comment claims this "restricts script sources to prevent XSS (CWE-79)"
script-src 'self' 'unsafe-eval' 'unsafe-inline';
```

Permitting both `unsafe-inline` and `unsafe-eval` negates the XSS protection the surrounding comment claims to provide. Any successful script injection — via a malicious prompt rendered unsafely, a compromised dependency, or a crafted model response — yields immediate exfiltration of the user's billing-capable API key. The in-code comment asserting CWE-79 mitigation is contradicted by the policy on the same line.

HIGH 30 known dependency vulnerabilities, 1 critical

`npm audit` against the committed lockfile (1,168 packages: 250 production, 882 dev):

SEVERITY	COUNT	REPRESENTATIVE ADVISORIES
CRITICAL	1	`tar` — arbitrary file creation/overwrite via hardlink path traversal
HIGH	22	`electron` ASAR integrity bypass; `builder-util-runtime` leaks `PRIVATE-TOKEN` / `Authorization` on cross-origin redirect; `axios` ReDoS; `next` DoS; `sharp`/libvips CVE-2026-33327/33328/35590/35591; `vite` path traversal; `postcss` XSS; `form-data` CRLF injection; `js-yaml` quadratic DoS; `tmp` path traversal
MODERATE	5	—
LOW	2	`@babel/core` arbitrary file read via `sourceMappingURL`

The Electron and electron-builder advisories are directly relevant because this project ships desktop installers built with that toolchain.

HIGH Unsigned binaries with documented Gatekeeper bypass

The build configuration disables code-signing verification on both desktop platforms:

```
"mac": { "gatekeeperAssess": false }
"win": { "signAndEditExecutable": false }
```

The README then instructs users to strip macOS quarantine attributes from the downloaded application:

```
$ xattr -cr "/Applications/Open Higgsfield AI.app"
```

This trains users to disable an operating-system integrity control on an unsigned binary obtained over the internet. Combined with the absence of any CI/CD pipeline (below), there is no verifiable chain of custody between the published source and the distributed installers.

HIGH No CI/CD, no release verification

The repository contains no `.github/workflows` directory. There is no automated build, test, lint or release pipeline for a project distributing signed-less desktop binaries to a 25k-star audience. Releases are produced on developer machines — three commits in the history are authored by `Ubuntu <ubuntu@ip-172-31-36-137.us-east-2.compute.internal>`, indicating commits made directly from an EC2 instance.

MEDIUM **Key-handling claim is imprecise in web mode**

The interface tells users: "*Your API key is stored locally and never sent anywhere except api.muapi.ai.*" This holds for the Electron renderer, which calls the upstream directly. In browser mode the code explicitly routes through the host application's server-side proxy to bypass CORS (`packages/studio/src/muapi.js`), so the key transits whichever server hosts the deployment. For a self-hosted instance that is the user's own server; for any shared or third-party deployment it is not. The blanket claim is inaccurate.

CLEAN **No secrets committed**

A regex scan across all commits and blobs in the full history — covering OpenAI-style keys, AWS access key IDs, GitHub tokens, Google API keys, Slack tokens and PEM private-key headers — returned **no matches**. No `.env`, credential, `.pem` or key files were ever added. The current tree contains no hardcoded API keys. This is a genuine positive and reflects correct `.gitignore` discipline.

MEDIUM **Content-policy positioning carries legal exposure**

The repository markets itself on removing safety controls — "Unrestricted", "Uncensored" and "No content filters" appear repeatedly across the README. Combined with image, video and **lip-sync** generation, an operator deploying this stack inherits meaningful regulatory exposure: non-consensual synthetic imagery, likeness rights, and jurisdiction-specific deepfake legislation. Any commercial adopter would need independent legal review and its own moderation layer, neither of which the project provides.

Architecture & code quality

MEDIUM **Three build systems and a duplicated UI layer**

The repository simultaneously carries Next.js (`next.config.mjs`, `app/`), Vite (`vite.config.mjs`, `index.html`, `src/`) and Electron (`electron/`) conventions. More significantly, the studio UI exists twice in parallel implementations — vanilla DOM modules under `src/components/*.js` and React components under `packages/studio/src/components/*.jsx`. `McpCliStudio`, for example, is implemented in both. Every feature change carries a latent two-place maintenance cost with no mechanism keeping the copies in sync.

SIGNAL	OBSERVED	ASSESSMENT
Test files	4, all local-inference scoped	Core studios, API proxies and key handling are untested
Largest source files	`ImageStudio.js` ~1,240 lines; `LipSyncStudio.js` ~670	Well past reasonable single-file limits
Repository weight	125 MB clone for 94 source files	Binary assets committed to history; no LFS
Manifest consistency	`"private": true` alongside `"license": "MIT"`	Contradictory; blocks publication while claiming open distribution

SIGNAL	OBSERVED	ASSESSMENT
README	638 lines / 40 KB	Predominantly marketing; SEO-oriented rather than technical
Licence file	MIT, clean, © 2026	Valid

Project health & governance

HIGH **Bus factor of one**

Commit attribution across 330 commits, after collapsing duplicate identities:

CONTRIBUTOR	IDENTITIES	COMMITS	SHARE
Anil Matcha	3	228	69.1%
Jaya Prasad Kavuru	2	70	21.2%
All others (15 people)	15	32	9.7%

Two people account for **90.3%** of all commits; the principal alone accounts for 69%. The 4,381 forks and 25k stars have produced 32 external commits. Community engagement is essentially nil relative to apparent popularity — consistent with the provenance finding that the social proof was inherited rather than earned.

Commit hygiene is weak: six commits are attributed to `Developer <dev@example.com>` and two to `Developer <developer@example.com>` — placeholder identities left unconfigured.

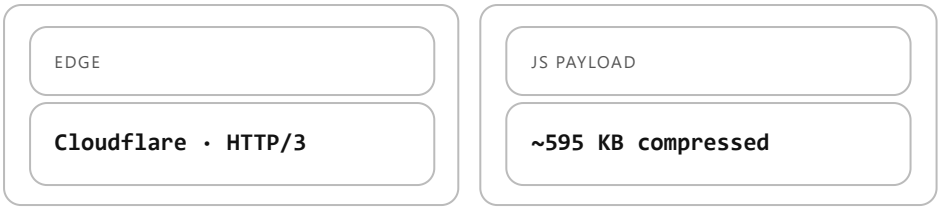
LOW **Development activity is real and current**

In fairness to the target: development is genuinely active. 330 commits over six months (Feb 25, Mar 39, Apr 76, May 74, Jun 28, Jul 88), 13 tags, last push on the day of assessment. The project is not abandoned. The concern is concentration and direction, not abandonment.

Target B — DASCA

APPLICATION	PUBLISHER	PRODUCT
app.dasca.studio	DASCA Studio	Browser trial

DESKTOP PRICE	STACK	REQUIRES
USD 39	Vite · React · Three.js	WebGL2



DASCA is a real-time visual-effects tool for images, video and 3D, delivered as a desktop application with a browser-based trial. The trial is not a marketing shell — it is a functioning creative environment. Interface inspection reveals:

- **Effect chain compositor** — stackable shader effects applied bottom-to-top, with a Bayer dithering effect and configurable matrix size present by default.
- **Node-based signal routing** — audio analysis exposing Bass, Mid, Treble, Level, Peak and Transient as connectable output ports, alongside generated Sine and Saw oscillators, each patchable into effect parameters.
- **GLSL shader playground** — a live code editor (CodeMirror, 516 KB bundle) for writing shaders directly.
- **Kinetic typography presets**, before/after comparison, canvas recording, zoom, aspect-ratio locking, undo/redo and export.

This is a technically ambitious product. The signal-graph architecture in particular is non-trivial engineering and is not typical of a thin web demo.

Security posture

STRONG Header and transport configuration

CONTROL	VALUE	ASSESSMENT
CSP `script-src`	`'self'` + Clarity + Cloudflare Insights	GOOD No `unsafe-inline`, no `unsafe-eval`
`frame-ancestors`	`'none'`	GOOD
`object-src` / `base-uri` / `form-action`	`'none'` / `'self'` / `'self'`	GOOD
`X-Frame-Options`	`DENY`	GOOD
`X-Content-Type-Options`	`nosniff`	GOOD
`Referrer-Policy`	`strict-origin-when-cross-origin`	GOOD
`Permissions-Policy`	camera, mic, geolocation, payment all `()`	GOOD
TLS protocols	1.2 and 1.3 only — 1.0/1.1 refused	GOOD
Certificate	Google Trust Services WE1, valid to 4 Sep 2026	GOOD
Sourcemaps	None exposed	GOOD
Secrets in bundles	None found	GOOD
`Strict-Transport-Security`	**Absent**	GAP

The CSP here is materially stronger than the one shipped by Target A, and notably avoids the `unsafe-inline` / `unsafe-eval` combination that undermines Target A's policy.

Findings

HIGH CSP blocks the site's own Google Analytics — zero data collected

The page loads Google Analytics property `G-2VJGNYXSNB`, but the CSP omits `googletagmanager.com` from `script-src` and provides no nonce or hash for the inline configuration block. Both the external script and the inline initialiser are blocked. Confirmed in an instrumented browser:

```
[ERROR] Loading the script 'https://www.googletagmanager.com/gtag/js?id=G-2VJGNYXSNB'
        violates the following Content Security Policy directive:
        "script-src 'self' https://www.clarity.ms ... " The action has been blocked.

[ERROR] Executing inline script violates ... 'script-src'. Either the 'unsafe-inline'
        keyword, a hash ('sha256-inDb5F090DgQjnxG7PRFANoj0D2UXdJi9tneItS2w5A='), or a nonce
        is required to enable inline execution. The action has been blocked.

Network: [GET] https://www.googletagmanager.com/gtag/js?id=G-2VJGNYXSNB => [FAILED] csp
```

Microsoft Clarity, which *is* allowlisted, functions normally (three successful `POST` requests to `v.clarity.ms/collect` returning 204). The practical effect is that any business decision made on the assumption that GA data exists for this property is unsupported. **Remediation:** add `https://www.googletagmanager.com` to `script-src` and `https://*.google-analytics.com` to `connect-src`, then attach the published hash `sha256-inDb5F090DgQjnxG7PRFANoj0D2UXdJi9tneItS2w5A=` to `script-src` for the inline block — or move the initialiser into the bundle.

MEDIUM No HSTS on either the app subdomain or the apex

`Strict-Transport-Security` is absent from `app.dasca.studio` and `dasca.studio`. Without it, a first visit over an attacker-controlled network can be downgraded to HTTP before the redirect completes. The domain is not in the HSTS preload list. **Remediation:** enable HSTS at the Cloudflare edge — `max-age=31536000; includeSubDomains; preload` — after verifying every subdomain serves HTTPS.

MEDIUM Session recording active without visible consent

Microsoft Clarity is loaded and transmitting on page load. Clarity captures session replay — pointer movement, clicks and interaction sequences. No consent banner or cookie notice appeared during assessment. For visitors in the EU/UK this engages GDPR and ePrivacy consent requirements, which are not satisfied by a privacy policy alone where replay is concerned. **Remediation:** gate Clarity initialisation behind explicit opt-in for applicable jurisdictions.

MEDIUM **DMARC set to monitor-only; SPF softfail**

RECORD	VALUE	ISSUE
SPF	`v=spf1 include:_spf.mx.cloudflare.net ~all`	`~all` softfail — unauthorised senders are marked, not rejected
DMARC	`v=DMARC1; p=none; rua=…@dmarc-reports.cloudflare.net`	`p=none` takes no action on failures

Reporting is correctly configured, so the groundwork is done. As it stands the domain remains practically spoofable for phishing that impersonates DASCA. **Remediation:** review aggregate reports, then progress `p=none` → `p=quarantine` → `p=reject` and tighten SPF to `-all`.

LOW **No security.txt, despite a misleading HTTP 200**

`/.well-known/security.txt` returns **200 OK** — but serves the SPA's `index.html`, because the single-page catch-all matches every unknown path. An automated scanner will record the file as present. There is no vulnerability-disclosure contact. **Remediation:** publish a real `security.txt` with a `Contact:` and `Expires:` field, and return proper 404s for unknown `/.well-known/` paths. (`sitemap.xml`, by contrast, is genuine.)

LOW **Wildcard CORS on the HTML document**

`Access-Control-Allow-Origin: *` is returned on the main document. For a public static page this carries no direct exploit path, but it is an unnecessarily permissive default. If any authenticated or user-specific route is later served from the same origin with this header inherited, it becomes a real issue. **Remediation:** scope the header to the asset paths that require it.

LOW **Shader compilation warnings indicate rendering-correctness risk**

The WebGL compiler emitted two warnings on load:

```
warning X3595: gradient instruction used in a loop with varying iteration;
               partial derivatives may have undefined value
warning X4000: use of potentially uninitialized variable
               (f_sampleWithChromaticAberration)
```

Both are genuine correctness concerns rather than style notes. Texture gradients inside variable-iteration loops produce driver-dependent results, and an uninitialised variable in the chromatic-aberration sampler may render differently — or produce artefacts — across GPU vendors. For a product whose entire value proposition is visual output fidelity, cross-GPU consistency is a commercial concern. **Remediation:** hoist gradient sampling out of the varying loop (or use explicit-LOD sampling), and initialise the flagged variable at declaration.

Performance

ASSET	UNCOMPRESSED	TRANSFERRED	NOTE
`index-C5Y5sJuT.js`	1,154 KB	306 KB	Application core
`vendor-codemirror-*.js`	516 KB	172 KB	Shader editor — candidate for lazy loading
`vendor-three-*.js`	471 KB	118 KB	WebGL engine — required at boot
`vendor-react-*.js`	11 KB	—	Thin; React largely inlined into core
Total JS	**~2,152 KB**	**~595 KB**	

Compression is working well (roughly 72% reduction). The load is nonetheless substantial for a trial experience intended to convert visitors. CodeMirror at 172 KB transferred is the clearest optimisation: the shader playground is behind a button, so the editor could be dynamically imported on first use rather than at boot. Vendor chunking is otherwise sensible, and asset hashing is correct for cache-busting.

Relationship analysis

A specific question in this engagement was whether DASCA is built on, forked from, or otherwise derived from Open-Generative-AI. **It is not.** No relationship of any kind was found.

DIMENSION	OPEN-GENERATIVE-AI	DASCA
Purpose	Generative AI media via remote models	Real-time GPU visual effects, local rendering
Stack	Next.js + Electron + Vite + vanilla DOM	Vite + React + Three.js + CodeMirror
Compute	Remote — `api.muapi.ai`	Client-side WebGL2
Model	Open-source funnel to paid API	Commercial desktop app, USD 39, free trial
Third-party calls	MuAPI, Vadoo	Clarity, Google Fonts, Cloudflare (GA blocked)
Attribution	Anil Matcha / SamurAIGPT	DASCA Studio

The only shared attributes are Cloudflare as edge provider and broad membership of the creative-media category. Bundle inspection found no MuAPI reference, no shared identifiers and no common code in DASCA.

Risk register

#	TARGET	RISK	SEVERITY	RECOMMENDED ACTION
1	Repo	Undisclosed commercial funnel to author-connected paid API	CRITICAL	Treat as vendor lock-in; price MuAPI per-generation cost into any adoption model
2	Repo	API key in localStorage + `unsafe-inline` / `unsafe-eval` CSP	CRITICAL	Do not deploy multi-user; move key server-side; remove unsafe CSP directives

#	TARGET	RISK	SEVERITY	RECOMMENDED ACTION
3	Repo	1 critical + 22 high dependency CVEs	HIGH	Full `npm audit fix` and Electron upgrade before any use
4	Repo	Unsigned binaries; documented Gatekeeper bypass	HIGH	Never distribute internally; build from source if adopting
5	Repo	Inherited stars misrepresent adoption	HIGH	Discount social proof entirely in evaluation
6	Repo	Bus factor 1 (69% single author)	HIGH	Assume no continuity guarantee; fork if depended upon
7	Repo	No CI/CD; untested core	HIGH	Assume no regression safety net
8	Repo	Unrestricted-content positioning; lip-sync + video	MEDIUM	Independent legal review; add moderation before any deployment
9	DASCA	GA fully blocked by own CSP	HIGH	Fix CSP; treat all historical GA data for this property as absent
10	DASCA	No HSTS on app or apex	MEDIUM	Enable at edge, then preload
11	DASCA	Session replay without consent gate	MEDIUM	Consent gate for EU/UK visitors
12	DASCA	DMARC `p=none`, SPF `~all`	MEDIUM	Escalate to `p=reject` / `-all`
13	DASCA	Shader warnings — cross-GPU output variance	LOW	Fix gradient-in-loop and uninitialised variable
14	DASCA	No security.txt behind misleading 200	LOW	Publish real file; 404 unknown well-known paths

Verdicts

VERDICT — OPEN-GENERATIVE-AI

Do not adopt as described

The project is not what its description claims. It is a competent front-end client for a commercial generation API, published as free self-hosted open-source software, with the commercial relationship undisclosed and the adoption signals inherited from a differently-named predecessor. On top of that positioning problem sit concrete technical defects: a critical key-handling weakness paired with a CSP that defeats itself, 23 critical/high dependency vulnerabilities, unsigned binaries accompanied by instructions to bypass macOS Gatekeeper, no CI/CD, and effectively no test coverage of the core application.

Narrow, conditional use. As a reference implementation for integrating MuAPI, or as a personal single-user tool on an isolated machine after a full dependency upgrade, it has some value. It should not be deployed multi-user, exposed to untrusted input, distributed as a binary internally, or represented to stakeholders as self-hosted or free. Do not treat 25k stars as diligence.

VERDICT — DASCA

Proceed, with routine remediation

DASCA is a credible, well-engineered product. The signal-routing graph, shader compositor and live GLSL environment represent real technical depth, and the security baseline — restrictive CSP without unsafe directives, modern TLS only, no exposed sourcemaps or secrets — is better than most independent web applications of comparable size.

Every finding against it is a configuration or process gap rather than a design flaw, and all are addressable within roughly a day of work. The analytics failure is the priority: it is silent, it has likely persisted since launch, and any growth or conversion decision informed by that GA property is resting on nothing. Fix the CSP, enable HSTS, gate session replay behind consent, escalate DMARC, and the posture becomes strong.

Evidence appendix

Every claim in this report traces to a reproducible observation. Key raw outputs:

A1 — PROVENANCE GAP

```
$ gh api repos/anil-matcha/open-generative-ai --jq '{created,pushed}'
{"created":"2023-05-09T14:12:37Z","pushed":"2026-07-28T08:52:33Z"}

$ git log --reverse --format='%ad | %s' --date=iso | head -1
2026-02-10 02:06:16 +0530 | Initial commit: Scaffold Open-Higgsfield-ai with Vite and README
```

A2 — MUAPI DEPENDENCY

```
$ grep -rhoE 'https?://[a-z.]+' --include=*.js app/ src/ components/ | sort | uniq -c | sort -rn
     16 https://api.muapi.ai
      4 https://github.com
      3 https://muapi.ai
      1 https://vadoo.tv

$ grep -rhoE 'process\.env\.[A-Z_]+' --include=*.js . | sort | uniq -c
      1 process.env.MUAPI_KEY   # the only env var in the codebase
```

A3 — SELF-DEALING LINK

```
$ gh api repos/SamurAIGPT/muapi-mcp-server --jq '{full_name,homepage}'
{"full_name":"SamurAIGPT/muapi-mcp-server",
 "homepage":"https://muapi.ai?utm_source=github&utm_medium=about&utm_campaign=muapi-mcp-server"}
```

A4 — DEPENDENCY AUDIT

```
$ npm audit --package-lock-only --json
VULN COUNTS: {"low":2,"moderate":5,"high":22,"critical":1,"total":30}
dependencies: {"prod":250,"dev":882,"optional":159,"peer":32,"total":1168}

CRITICAL | tar    | node-tar Vulnerable to Arbitrary File Creation/Overwrite via Hardlink Path Traversal
HIGH     | electron | Electron has ASAR Integrity Bypass via resource modification
HIGH     | builder-util-runtime | electron-updater: Cross-origin redirect leaks PRIVATE-TOKEN …
```

A5 — SECRET SCAN (CLEAN)

```
$ git log --all --format='%H' | while read c; do git grep -IEn \
    '(sk-[A-Za-z0-9]{16,}|AKIA[0-9A-Z]{16}|ghp_…|AIza…|-----BEGIN (RSA|PRIVATE))' $c; done
>>> no matches across all 330 commits
```

B1 — DASCA CSP BLOCKING ITS OWN ANALYTICS

```
Browser console, https://app.dasca.studio
[ERROR] Loading the script 'https://www.googletagmanager.com/gtag/js?id=G-2VJGNYXSNB'
        violates … "script-src 'self' https://www.clarity.ms …". Blocked.
[ERROR] Executing inline script violates … Blocked.

Network
10. [GET]  https://www.googletagmanager.com/gtag/js?id=G-2VJGNYXSNB => [FAILED] csp
22. [POST] https://v.clarity.ms/collect                              => [204]
25. [POST] https://v.clarity.ms/collect                              => [204]
```

B2 — TLS NEGOTIATION

```
$ for v in tls1 tls1_1 tls1_2 tls1_3; do openssl s_client -$v -connect app.dasca.studio:443 …
tls1   -> no protocols available   # correctly refused
tls1_1 -> no protocols available   # correctly refused
tls1_2 -> ECDHE-ECDSA-CHACHA20-POLY1305
tls1_3 -> TLS_AES_256_GCM_SHA384

issuer=C=US, O=Google Trust Services, CN=WE1
notBefore=Jun 5 2026  notAfter=Sep 4 2026
```

B3 — MISSING HSTS

```
$ curl -sSI https://app.dasca.studio | grep -i strict-transport
>>> NO HSTS
$ curl -sSI https://dasca.studio | grep -i strict-transport
>>> NO HSTS on apex either
```

B4 — EMAIL AUTHENTICATION

```
$ nslookup -type=TXT dasca.studio
"v=spf1 include:_spf.mx.cloudflare.net ~all"
$ nslookup -type=TXT _dmarc.dasca.studio
"v=DMARC1; p=none; rua=mailto:…@dmarc-reports.cloudflare.net"
```

Methodology & limitations. Findings reflect the state of both targets on 29 July 2026. Target A was assessed by static analysis of a full-history clone; the application was not executed, so runtime behaviour is inferred from source. Target B was assessed passively — no active security testing was performed, so this report cannot speak to authentication, authorisation, injection resistance or server-side logic, and the absence of such findings must not be read as their absence in the product. Dependency vulnerability counts derive from the committed lockfile and reflect advisory data at the time of assessment. Attribution of MuAPI to the repository author is based on publicly observable organisation ownership, submodule targets and campaign-tagged marketing links; it is a strong inference from public evidence rather than a confirmed corporate record.

Independent technical due diligence · 29 July 2026 · Passive assessment