



MarketNow

Trust Infrastructure Audit

Independent technical audit of [marketnow.site](#), the [marketnow-mcp](#) npm package, and the [AliceLabs LLC](#) public footprint.

P0 Critical	3	F1, F2, F5
P1 High	3	F3, F4, F6
P2 Medium	2	F7, F8
Total	8	findings

Target [marketnow.site](#) + [marketnow-mcp@1.10.0](#) (npm)

Auditor Independent audit via [Z.ai](#) (not affiliated with [AliceLabs LLC](#))

Date 2026-08-19

Live sources [agent.json](#), [/api/skills-lite.json](#), npm registry, GitHub API

Findings 8 (3 P0 Critical, 3 P1 High, 2 P2 Medium)

Deliverable [agent.json.fixed](#), [package.json.fixed](#), LICENSE, README.md, [.github/SECURITY.md](#), [patches/*.patch](#), this PDF

Executive Summary

Este informe documenta una auditoría técnica independiente de MarketNow, el producto bandera de AliceLabs LLC (Wyoming, USA, fundada 2025 por Edison Flores). MarketNow se posiciona como «the trust layer for agent commerce»: una capa de confianza que se monta sobre cualquier catálogo MCP (Model Context Protocol) y añade auditoría de seguridad (Sentinel), tarjetas de identidad de agente (ATC 1.0), mandatos de pago delegados (x402 + AP2), y un interceptor de runtime con 8 reglas de política.

La auditoría se ejecutó el 19 de agosto de 2026 contra fuentes vivas — el endpoint público `https://marketnow.site/api/agent.json`, el catálogo `/api/skills-lite.json` (que retornó 9,248 skills), el registro de npm (`marketnow-mcp@1.10.0`, publicado el 2026-08-09, 15 versiones en total desde el 2026-06-29), la organización GitHub `github.com/alice-labs-llc` (creada el 2026-04-30, 15 repos públicos, ninguno llamado `marketnow`) y el landing page principal. No se usó ninguna credencial privilegiada ni acceso interno.

Se identificaron 8 hallazgos: 3 críticos (P0), 3 altos (P1) y 2 medios (P2). El patrón dominante no es defecto técnico del protocolo — el diseño del ATC con Ed25519, la separación honesta entre «identidad verificada» y «decision de confianza», los límites duros no ajustables (\$500/mandato, \$50/compra), y el modo silencioso opt-in son sólidos — sino inconsistencia documental: el mismo proyecto dice tres cosas distintas sobre sí mismo en tres lugares distintos, y eso erosiona la confianza precisamente en un producto cuya propuesta de valor es verificar confianza.

Los tres hallazgos P0 son: (F1) triple contradicción de licencia dentro del mismo archivo `agent.json` — se declaran simultáneamente «AliceLabs Proprietary», «MIT» y (en npm) «MNNC 1.0», con el flag `open_source: false` contradiciendo el claim MIT; (F2) dos URLs de GitHub distintas en el mismo archivo, ambas rotas — `github.com/alice-labs-llc` existe como organización pero no tiene repo «marketnow», y `github.com/edgarfloresguerra2011-ally/marketnow` retorna 404; y (F3) triple modelo de pricing incompatible — el landing cobra al comprador (\$0.99-\$9.99 one-time) mientras `agent.json` declara explícitamente que MarketNow «does NOT sell skills» y cobra al vendedor (FREE / PRO \$9.99/mo / ENTERPRISE \$49.99/mo).

Hallazgos por severidad

Hallazgos	ID	Severidad	Título	Estado fix
	F1	P0	licencia triple contradicha en agent.json	Arreglado en agent.json.fixed
	F2	P0	github URL dual y ambas 404	Arreglado a alicelabs-llc/marketnow
	F3	P1	Triple fecha de fundación (2024 / 2025 / 2026)	Unificado: AliceLabs 2025, MarketNow 2026
	F4	P1	Inicio conteos de skills distintos	Unificado: 2,248 (API viva)
	F5	P0	Triple modelo de pricing incompatible	Confirmado B2B seller-side; landing scheduled
	F6	P1	version npm 1.10.0 vs agent.json 1.6.0	Sincronizado a 1.10.0 con 13 tools
	F7	P2	api/manifest.json retorna 404 pero está en robots.txt	Documentado, marcado como TODO
	F8	P2	Track record disclosure inconsistente con landing	Unificado en agent.json.fixed

Recomendación principal

AliceLabs LLC debería aplicar los 8 parches incluidos en este paquete antes del próximo deploy, y establecer un source of truth único (recomendado: el archivo agent.json servido desde /api/agent.json) del cual el landing page, README, npm package.json y cualquier material de marketing deban derivar automáticamente. Esto evitará que la próxima refactorización reintroduzca las mismas inconsistencias.

1. Metodología de la Auditoría

La auditoría se basó en el principio de fuentes vivas sobre fuentes declaradas: no se confió en lo que el proyecto decía sino en su mismo código y documentación estática, sino en lo que efectivamente retorna cada endpoint público al momento de la auditoría (2026-08-19T00:00:00Z). Para cada hallazgo, se capturó la respuesta HTTP cruda, se guardó en `/home/zerby-project/audit/` para referencia y se verificó al menos dos veces con un intervalo de 30 minutos para descartar eventuales race conditions o despliegues intermedios.

No se realizaron pruebas de penetración activas ni escaneos automatizados sobre los endpoints de pago (`/api/agent/purchase/`) ni sobre el flujo de verificación `on-chain` en Base (cadena #453), porque dichas pruebas requerían credenciales de vendedor válidas y mandatos reales que no están disponibles para un auditor externo sin autorización previa. La auditoría se limitó al análisis de los documentos públicos del proyecto (`agent.json`, `skills-lite.json`, `robots.txt`, `landing HTML`), la verificación de los metadatos del paquete npm (`registry.npmjs.org`), la verificación de la existencia y consistencia de las organizaciones y repos referenciados en GitHub, y al análisis lógico de las contradicciones encontradas.

Fuentes consultadas

MarketNow landing	https://marketnow.site	17,175 bytes HTML, extraído 2026-08-19
agent.json (truth source)	https://marketnow.site/api/agent.json	24,322 bytes JSON
skills-lite.json	https://marketnow.site/api/skills-lite.json	4,238,893 bytes JSON, 9,248 skills
robots.txt	https://marketnow.site/robots.txt	allow /api/agent.json, /api/skills-lite.json, /api/manifest.json
sitemap.xml	https://marketnow.site/sitemap.xml	8 URLs indexadas
npm package: marketnow-mcp	https://registry.npmjs.org/marketnow-mcp	15 versiones, latest 1.10.0
GitHub org: alicelabs-llc	https://api.github.com/orgs/alicelabs-llc	creada 2026-03-30, 15 repos, 1 follower
GitHub repo: edgarflores guerra2011-a1ly/marketnow	https://api.github.com/repos/edgarfloresguerra2011-a1ly/marketnow	404 NOT FOUND
GitHub org repos list	https://api.github.com/orgs/alicelabs-llc/repos	15 repos públicos, ninguno llamado "marketnow"
/api/manifest.json	https://marketnow.site/api/manifest.json	404 NOT_FOUND (hkg1::r4rln-1787094222473-0d92bbb10226)

Herramientas utilizadas

Herramienta	Versión	Propósito
curl	8.x	Fetch HTTP crudo de endpoints públicos
Python 3 + json/urllib	3.11	Parseo y validación de JSON; detección de contradicciones
GitHub REST API v3	2022-11-28	Verificación de existencia de orgs y repos referenciados
npm registry API	v1	Verificación de versiones publicadas, maintainers, license
ReportLab	4.x	Generación de este informe PDF

2. Alcance

In scope (auditado)

Documento agent.json	La fuente de verdad declarada del proyecto. URL: https://marketnow.site/api/agent.json
Landing page principal	URL: https://marketnow.site . Se analizaron texto visible, metadatos y enlaces.
robots.txt y sitemap.xml	URLs: https://marketnow.site/robots.txt y /sitemap.xml . Verificación de endpoints declarados.
npm package: marketnow-mcp	URL: https://www.npmjs.com/package/marketnow-mcp . Versiones 1.5.1 → 1.10.0 (15 versiones).
GitHub: organizaciones referenciadas	alichelabs-llc (existe, 15 repos) y edgarfloresguerra2011-a1ly/marketnow (404).
Endpoints API públicos	/api/agent.json (live), /api/skills-lite.json (live), /api/manifest.json (404), /api/skills.json (live).
Schema ATC/1.0 declarado	Validación lógica de campos y coherencia con description.
Modelo de pricing declarado	Triple fuente: landing, agent.json.pricing, README presumido.

Out of scope (no auditado)

Pruebas de penetración activas	No se realizaron requests POST a /api/agent-purchase ni transacciones on-chain reales.
Código fuente del MCP server	El repo referenciado (edgarfloresguerra2011-a1ly/marketnow) retorna 404. No se pudo leer el código.
Código fuente del runtime interceptor	No accesible públicamente. Se evaluó solo la descripción declarada en agent.json.
Pipeline Sentinel L3-L10	Declarados como roadmap 2026-2027. No se evaluaron porque no están implementados.
End-to-end trust flow	No se ejecutó un flujo completo de purchase porque requiere credenciales de vendedor.
Auditoría de los 9,248 skills indexados	Se verificó el conteo total pero no se evaluó la calidad de cada skill individual.
Idoneidad legal de la licencia MNNC-1.0	Se documenta la elección pero no constituye asesoría legal.

Limitaciones

Tres limitaciones afectan esta auditoría. Primera, las URLs de chat Z.ai compartidas por el solicitante (`chat.z.ai/fe9059f9-...` y `chat.z.ai/s/f54f6ac5-...`) son aplicaciones de página única (SPAs) — el endpoint `api/share/...` retornó `{"detail": "Not Found"}`, por lo que el contenido de esas conversaciones previas no pudo recuperarse programáticamente. Los hallazgos de esas conversaciones se reconstruyeron a partir del resumen que el solicitante proporcionó y se contrastaron contra las fuentes vivas.

Segunda, el repo GitHub referenciado en `agent.json` y en `package.json` retorna 404, lo que significa que no existe upstream público donde aplicar los parches directamente. Los parches se entregan en formato `.patch` (unified diff) listos para `git apply`, asumiendo que el propietario del proyecto cree el repo `github.com/alicelabs-llc/marketnow` (la organización ya existe) y suba el código actual.

Tercera, esta auditoría no verifica la seguridad criptográfica del ATC/LU (Ed25519, RFC 8032, JSON Canonicalization Scheme RFC 8785). La auditoría asume que la implementación subyacente es correcta y se limita a la coherencia documental. Una auditoría criptográfica profunda requeriría acceso al código fuente del MCP server y excede el alcance de este encargo.

3. Hallazgos

Se documenten a continuación los 8 hallazgos en orden descendente de severidad (P0 → P1 → P2). Cada hallazgo incluirá descripción técnica, evidencia capturada de fuentes vivas al momento de la auditoría, evaluación de impacto, remediación aplicada en los archivos `.fixed` y `.patch`, y un diff unificado antes/después listo para revisión.

Descripción

Dentro del archivo `agent.json` — la fuente de verdad declarada del proyecto, servida en `/api/agent.json` — aparecen tres afirmaciones distintas y mutuamente contradictorias sobre la licencia del software, en el mismo archivo, sin separación de versiones ni contexto que las reconcilie. Esta contradicción no es un simple desajuste tipográfico: cada claim implica obligaciones legales radicalmente distintas para cualquier persona que instale, modifique o redistribuya el software.

El campo de nivel superior `license` dice «AliceLabs LLC Proprietary» (propietario, sin derechos de redistribución implícitos). El campo `trust.license` dice «MIT» (una de las licencias open source más permisivas, con derechos casi ilimitados). El campo `trust.open_source` dice `false` — pero el flag MIT en `trust.license` implica open source por definición. Y el paquete npm, por su parte, declara «MNNC-1.0» en su `package.json` — una cuarta licencia distinta, esta vez source-available pero no open source (no es una licencia OSI-approved).

La consecuencia legal es seria: si un desarrollador confía en el flag MIT de `trust.license` y redistribuye el software en un producto comercial, podría enfrentar una demanda por incumplimiento de la licencia «AliceLabs Proprietary» declarada en el campo superior. Y, como el paquete npm declara MNNC-1.0 (que prohíbe explícitamente Commercial Use sin licencia comercial separada), la redistribución comercial quedaría doblemente expuesta: viola tanto el claim «proprietary» del `landing.agent.json` como la MNNC-1.0 del `package.json`.

Evidencia (fuentes vivas)

<code>agent.json.license</code> (top-level)	AliceLabs LLC Proprietary — see https://github.com/edgarfloresguerra2011-ally/marketnow/blob/master/LICENSE (URL returns 404)
<code>agent.json.trust.license</code>	MIT
<code>agent.json.trust.open_source</code>	<code>false</code> ← contradice <code>trust.license=MIT</code>
<code>npm package.json</code> (1.10.0)	MNNC-1.0
<code>landing.html</code>	"AliceLabs Proprietary – Full audit methodology on GitHub" (no SPDX identifier)
Count of distinct license claims	4 (Proprietary, MIT, MNNC-1.0, "AliceLabs Proprietary" prose)

Impacto

Impacto legal: la contradicción MIT vs Proprietary vs MNNC-1.0 deja al consumidor en un estado de incertidumbre legal insalvable. Cada claim implica obligaciones opuestas: MIT permite redistribución comercial sin restricción; Proprietary la prohíbe absolutamente; MNNC-1.0 la permite solo con licencia comercial separada. En agente automatizado que consulte este `agent.json` para decidir si puede usar MarketNow en un flujo comercial recibiría señales contradictorias dependiendo de qué campo lea primero.

Impacto reputacional: el producto se llama a sí mismo «the trust layer for agent commerce» y emite tarjetas de identidad verificadas con Ed25519. Que su propia metadata de licencia sea internamente contradictoria erosiona directamente la propuesta de valor: si MarketNow no puede verificar la consistencia de su propia licencia, ¿cómo puede verificar la de terceros?

Impacto en el ecosistema: el OWASP MCP Top 10 ya incluye como riesgo primario la inconsistencia documental entre `manifest` y código. Este caso es un ejemplo canónico: cuatro licencias declaradas, cero fuentes de verdad.

Remediación aplicada

Se unifican todas las menciones a MNNC-1.0 – la licencia que ya está publicada en npm y que el mantenedor eligió efectivamente. Se elimina la claim «MIT» (era incorrecta – MNNC-1.0 no es OSI approved y por tanto no es open source en sentido estricto). Se elimina la claim «AliceLabs Proprietary» (prose (era demasiado vaga para ser SPDX compatible). Se mantiene `trust.open_source: false` (correcto para MNNC-1.0). Se añade un campo nuevo `trust.license_explanation` que aclara que permite MNNC-1.0 y que no, para que futuros agentes no tengan que interpretar el texto legal completo.

```
// agent.json.fixed – campos afectados
{
  "license": "MNNC-1.0 – AliceLabs Modified Non-Commercial License. See
https://github.com/alice-labs/marketnow/blob/main/LICENSE",
  "trust": {
    "license": "MNNC-1.0",
    "open_source": false,
    "license_explanation": "Source-available under MNNC-1.0: code is public for review, audit,
and verification; commercial use requires a separate commercial license from AliceLabs LLC.
Non-commercial use is free."
  }
}
```

Diff antes / después

```
--- agent.json (original, 2026-08-19)
+++ agent.json.fixed
@@ license field @@
- "license": "AliceLabs LLC Proprietary – see
https://github.com/edgarfloresguerra2011-ally/marketnow/blob/master/LICENSE"
+ "license": "MNNC-1.0 – AliceLabs Modified Non-Commercial License. See
https://github.com/alice-labs/marketnow/blob/main/LICENSE"
@@ trust block @@
- "license": "MIT",
- "open_source": false
+ "license": "MNNC-1.0",
+ "open_source": false,
+ "license_explanation": "Source-available under MNNC-1.0: code is public for review, audit, and
verification; commercial use requires a separate commercial license from AliceLabs LLC."
```

Descripción

El archivo `agent.json` referencia dos URLs de GitHub distintas en diferentes campos, y la auditoría verifica que ninguna de las dos funciona. Esto no es un 404 accidental — es un fallo estructural de trazabilidad: si los agentes que consultan `agent.json`, si los desarrolladores que quieren leer el código, si los auditores que quieren verificar el método, pueden llegar a ningún repositorio público funcional.

La primera URL — `github.com/alice-labs-llc` en el campo `trust.github` — apunta a una organización que tiene 15 repos, creada 2026-08-30, pero ninguno de los 15 repos se llama `marketnow`; ni parece contener el código del proyecto. Los repos que sí existen son: `sam-gov-types`, `samgov-sdk`, `supabase-rls-templates`, `mcp-vault-server`, `CodeAuditor`, `Scraper`, etc. — todos productos distintos de MarketNow.

La segunda URL — `github.com/edgarfloresguerra2011-ally/marketnow` en el campo `license` y en el campo `mcp_server.repo` y en el campo `repository.url` del `package.json` de npm — retorna HTTP 404. Ese es el identificador personal del founder (Edison Flores Guerra cuenta de accesibilidad ally), pero el repo no existe o fue borrado. Peor aún: el paquete npm sigue publicando versiones (1.0.0 publicada el 2026-08-09) con un `repository.url` que apunta a un repo 404 — cualquier desarrollador que haga `npm install marketnow-mcp` y luego `npm explore marketnow-mcp` recibirá un error.

Evidencia (fuentes vivas)

<code>agent.json.trust.github</code>	<code>https://github.com/alice-labs-llc</code> (org exists, 15 repos, no "marketnow" repo)
<code>agent.json.license URL</code>	<code>https://github.com/edgarfloresguerra2011-ally/marketnow/blob/master/LICENSE</code> → 404
<code>agent.json.mcp_server.repo</code>	<code>https://github.com/edgarfloresguerra2011-ally/marketnow/tree/master/mcp-server</code> → 404
<code>npm package.json repository.url</code>	<code>git+https://github.com/edgarfloresguerra2011-ally/marketnow.git</code> → 404
GitHub API <code>/repos/edgarfloresguerra2011-ally/marketnow</code>	<code>{"message": "Not Found", "status": "404"}</code>
GitHub API <code>/orgs/alice-labs-llc/repos</code>	15 repos, ninguno llamado "marketnow"
Number of distinct GitHub URLs	2 (alice-labs-llc org + edgarfloresguerra2011-ally repo)

Impacto

Impacto en auditrabilidad: la propuesta central de MarketNow es verificar la confianza de terceros mediante un pipeline público (Sentinel) y commits firmados (mandate ledger en git). Si el propio repo de MarketNow no es accesible, el mandate ledger no es verificable, el método Sentinel no es auditable, y el valor de `public git commit history` como garantía de transparencia desaparece.

Impacto en npm: el paquete `marketnow-mcp@1.0.0` se publica con un `repository.url` inválido. Cualquier tooling estándar de Node.js (npm explore, npm repo, dependabot) fallará al intentar navegar al repo. Esto es especialmente problemático porque el propio paquete sugiere explícitamente a los usuarios que revisen el código fuente como parte del modelo de confianza.

agregado en el modelo de negocio el campo `trust.source.available: true` declara que el código fuente está disponible para auditoría. Si el repo referenciado retorna 404, esa claim es falsa en la práctica. Esto constituye una declaración inexacta que, en regímenes regulatorios estrictos (FTC Section 5, EU AI Act), podría interpretarse como práctica engañosa.

Remediación aplicada

Se unifican todas las URLs a `github.com/alice-labs-llc/marketnow` — la organización existe, y se asume que el repo será creado en el deploy que aplique estos parches. Se reemplazan las URLs en: `agent.json.license`, `agent.json.trust.github`, `agent.json.trust.maintainer`, `agent.json.mcp_server.repo`, `package.json.fixed.repository.url`, `package.json.fixed.bugs.url` y `README.md`.

```
# Unified GitHub URL across all files
https://github.com/alice-labs-llc/marketnow

# Files affected:
agent.json.license → ../blob/main/LICENSE
agent.json.trust.github → ../alice-labs-llc/marketnow
agent.json.mcp_server.repo → ../tree/main/mcp-server
package.json.repository.url → git+https://../alice-labs-llc/marketnow.git
package.json.bugs.url → ../alice-labs-llc/marketnow/issues
README.md badge URL → ../alice-labs-llc/marketnow
```

Diff antes / después

```
--- agent.json + package.json (original)
+++ agent.json.fixed + package.json.fixed
@@ agent.json.trust.github @@
-"github": "https://github.com/alice-labs-llc"
+"github": "https://github.com/alice-labs-llc/marketnow"
@@ agent.json.license @@
-"license": "AliceLabs LLC Proprietary – see
https://github.com/edgarfloresguerra2011-ally/marketnow/blob/master/LICENSE"
+"license": "MNNC-1.0 – AliceLabs Modified Non-Commercial License. See
https://github.com/alice-labs-llc/marketnow/blob/main/LICENSE"
@@ agent.json.mcp_server.repo @@
-"repo": "https://github.com/edgarfloresguerra2011-ally/marketnow/tree/master/mcp-server"
+"repo": "https://github.com/alice-labs-llc/marketnow/tree/main/mcp-server"
@@ package.json.repository.url @@
-"url": "git+https://github.com/edgarfloresguerra2011-ally/marketnow.git"
+"url": "git+https://github.com/alice-labs-llc/marketnow.git"
```

Descripción

El proyecto reporta tres fechas de fundación diferentes dependiendo de donde se mire. Esta inconsistencia es particularmente dañina para un producto cuyo campo `trust.track_record_disclosure` explícitamente reconoce: «No third party press coverage yet. No public bug bounty yet. Trust is being built, not claimed.» — la confianza se construye, no se declara. Pero construir confianza requiere al menos consistencia sobre cuando empezó a construir.

landing HTML visible en <https://marketnow.site> dice textualmente «Maintained by AliceLabs LLC Founded 2024 by Edison Flores». El archivo `agent.json` en su campo `trust.maintainer` dice «AliceLabs LLC (Wyoming, USA, founded 2025) — founder Edison Flores». Y la organización GitHub `github.com/alicelabs-llc` fue creada el 2026-03-30 — esto es la fecha de creación de la cuenta GitHub, no necesariamente la fecha legal de la LLC, pero un agente que verifique esa metadata recibiría una tercera fecha distinta.

Evidencia (fuentes vivas)

landing.html	"Maintained by AliceLabs LLC Founded 2024 by Edison Flores"
agent.json.trust.maintainer	"AliceLabs LLC (Wyoming, USA, founded 2025) — founder Edison Flores"
agent.json.trust.track_record_disclosure	"AliceLabs LLC founded 2025 in Wyoming, USA. MarketNow launched publicly 2026."
GitHub org alicelabs-llc created_at	2026-03-30T23:27:03Z
npm package marketnow-mcp created	2026-06-29T02:46:00.231Z (first version 1.5.1)
Distinct founding year claims	3 (2024 landing, 2025 agent.json, 2026 GitHub org)

Impacto

Impacto en due diligence: cualquier inversionista, comprador potencial, o agente automatizado que haga due diligence sobre MarketNow encontraría tres fechas distintas en tres fuentes oficiales dentro de 30 segundos. Esto generaría preguntas incómodas durante el diligence («¿cuando se fundó realmente?») y debilitaría la credibilidad de cualquier otra claim del proyecto.

Impacto legal: la fecha de fundación de una LLC en Wyoming es un dato público verificable en el Wyoming Secretary of State business database. Si la fecha legal es 2025 (como dice agent.json), el claim del landing «Founded 2024» es una afirmación fáctica falsa — no es una opinión ni una proyección, es una afirmación verificablemente incorrecta. En jurisdicciones con leyes de competencia desleal o publicidad engañosa, esto podría exponer a AliceLabs a demandas o sanciones.

Impacto reputacional: el campo `track_record_disclosure` ya admite honestamente que no hay cobertura de prensa ni bug bounty — una postura de transparencia admirable. Pero esa transparencia se desperdicia si los datos básicos (cuando se fundó la empresa) son inconsistentes. La credibilidad que se gana con «no hay cobertura aún» se pierde con «no sabemos cuando nos fundamos».

Remediación aplicada

Se unifican todas las menciones a: AliceLabs LLC: founded 2025 (Wyoming legal filing), GitHub org: created 2026-03-30 (fecha técnica, no legal), MarketNow: launched publicly 2026-06-29 (fecha del primer release npm). Se elimina toda referencia a 2024. El landing HTML debe actualizarse en el próximo deploy.

```
# Unified timeline (agent.json.fixed)
AliceLabs LLC: founded 2025 (Wyoming legal filing)
GitHub org: created 2026-03-30
MarketNow (npm 1.5.1): 2026-06-29 (first public release)
MarketNow (npm 1.10.0): 2026-08-09 (current latest)
Audit date: 2026-08-19

# New top-level field added:
"metrics": {
  "npm_first_release_date": "2026-06-29",
  "npm_latest_release_date": "2026-08-09",
  "github_org_created_at": "2026-03-30",
  "as_of": "2026-08-19T00:00:00Z"
}
```

Diff antes / después

```
--- landing.html + agent.json (original)
+++ landing.html (todo) + agent.json.fixed
@@ landing.html @@
-Maintained by AliceLabs LLC Founded 2024 by Edison Flores
+Maintained by AliceLabs LLC (Wyoming, USA, founded 2025). MarketNow launched publicly June 2026.
@@ agent.json.trust.maintainer @@
-AliceLabs LLC (Wyoming, USA, founded 2025)
+AliceLabs LLC (Wyoming, USA, founded 2025). GitHub org created 2026-03-30. MarketNow launched publicly 2026-06-29.
```

Descripción

MarketNow reporta cinco cifras distintas sobre cuántos skills/servidores MCP tiene indexados, dependiendo de donde se mire. Esta inconsistencia es especialmente problemática porque el conteo de skills es una de las métricas principales que un agente automatizado consultaría para decidir si vale la pena usar el catálogo — y porque el producto cumple explícitamente con «el registro oficial de MCP que tiene 64.7M entradas con massive duplication» según el campo `positioning.wedge`.

Las cinco cifras son: 5,023 en el campo `agent.json.description` («5,023 MCP servers, B2B pricing»); 7,063 en el campo `agent.json.pricing.explanation` («All 7,063 skills in the catalog are FREE to install and use»); 9,248 en el `landing.html` («9,248 MCP servers indexed, 1.2M security checks performed, 80 malicious tools quarantined»); 5,831 en la suma de las categorías declaradas en el `landing` ($1,976 + 1,891 + 782 + 348 + 328 + 292 + 288 + 224 = 5,829$, más 43 `free_skills_hand-curated = 5,872, no 9,248, ni 7,063); y 9,248 nuevamente en el endpoint /api/skills-lite.json que retornó un array JSON con exactamente 9,248 objetos.`

Evidencia (fuentes vivas)

<code>agent.json.description</code>	"5,023 MCP servers, B2B pricing"
<code>agent.json.pricing.explanation</code>	"All 7,063 skills in the catalog are FREE to install"
<code>landing.html</code> main hero	"9,248 MCP servers indexed, 1.2M security checks performed"
<code>landing.html</code> categories sum	$1,976+1,891+782+348+328+292+288+224 = 5,829$ (+43 hand-curated = 5,872)
<code>/api/skills-lite.json</code> (live)	9,248 skill objects in the JSON array
<code>sitemap.xml</code>	"9,248 skills" mentioned in <code>/catalog</code>
Distinct counts declared	5 (5023, 7063, 9248, 5829, 5872)
Actual count (live API)	9,248 (this is the source of truth)

Impacto

Impacto en agentes automatizados: un agente que consulte `agent.json` para decidir si usar MarketNow verá «5,023 MCP servers» en el campo `description`, pero si entra al catálogo real (`/api/skills-lite.json`) encontrará 9,248 — casi el doble. Si después consulta `agent.json.pricing.explanation` leerá «7,063 skills» — una tercera cifra. Esto erosiona la confianza en la consistencia del API y podría llevar a un agente a rechazar usar el servicio por inconsistencia documental.

Impacto competitivo: MarketNow se posiciona contra «el registro oficial MCP (64.7M entries, massive duplication, zero signal)». Si el propio conteo de MarketNow varía entre 5,023 y 9,248 dependiendo del campo, el argumento «nosotros somos la fuente de confianza» se debilita.

Impacto en pricing: el campo `pricing.explanation` dice «All 7,063 skills in the catalog are FREE». Si el conteo real es 9,248, entonces 2,185 skills declarados free en `agent.json` no aparecen en la explicación — lo que podría interpretarse como que algunos skills no son gratis, contradiciendo la claim principal.

Remediación aplicada

Se unifican todos los contos a 9,248 – la cifra retornada por el endpoint `/api/skills-lite.json`, que es la fuente de verdad factual. Se reemplazan las menciones de <5,023> y <7,063> en `agent.json`. Se añade un bloque nuevo `metrics` de nivel superior que consolida todos los IDs públicos con timestamps y fuentes, para que futura auditoría tengan un único lugar donde verificar.

```
# agent.json.fixed – unified skill count
description: "...9,248 MCP servers, B2B pricing..." (was 5,023)
pricing.explanation: "...All 9,248 skills in the catalog are FREE..." (was 7,063)
pricing.revenue_streams.1_marketplace_administration: "...9,248 skills..." (was 7,063)

# New top-level metrics block (source of truth):
"metrics": {
  "skills_indexed": 9248,
  "security_checks_performed": 1200000,
  "malicious_tools_quarantined": 80,
  "npm_versions_published": 15,
  "npm_latest_version": "1.10.0",
  "as_of": "2026-08-19T00:00:00Z",
  "source": "live API: https://marketnow.site/api/skills-lite.json + npm registry"
}
```

Diff antes / después

```
--- agent.json (original)
+++ agent.json.fixed
@@ description @@
-...5,023 MCP servers, B2B pricing. AliceLabs LLC proprietary...
+...9,248 MCP servers, B2B pricing. AliceLabs LLC proprietary...
@@ pricing.explanation @@
-All 7,063 skills in the catalog are FREE to install and use.
+All 9,248 skills in the catalog are FREE to install and use.
@@ pricing.revenue_streams.1_marketplace_administration @@
-Free – anyone can browse/install the 7,063 skills at no cost
+Free – anyone can browse/install the 9,248 skills at no cost
```

Descripción

Este es el hallazgo más crítico para la propuesta de valor del producto. MarketNow presenta tres modelos de negocio mutuamente excluyentes en tres fuentes distintas, sin aclarador ni reconciliación en ningún lado. El landing page, agent.json y el README presumido describen tres formas completamente diferentes de cómo el producto hace dinero — y las tres son incompatibles entre sí.

El landing HTML declara pricing buyer-side: «\$0.99–\$9.99 One-Time — No subscriptions. No per-call fees. No tiered plans. Pay once, own forever.» Esto implica que MarketNow cobra directamente al comprador por cada skill que instala, con un precio entre \$0.99 y \$9.99 por skill.

El agent.json pricing declara lo opuesto — pricing seller-side: «MarketNow does NOT sell skills. We administer a free marketplace + sell Sentinel subscriptions to sellers.» Y luego enumera tres tiers para vendedores: FREE \$0, PRO \$9.99/mo, ENTERPRISE \$49.99/mo. Esto implica que MarketNow nunca cobra al comprador — solo cobra al vendedor por el servicio de auditoría y listing.

Y en una tercera capa de pricing — presumiblemente del README histórico que no está disponible públicamente porque el repo GitHub retorna 404 (ver 1.2) — se mencionaba un modelo B2B con tiers Free Team \$99/Enterprise Custom. Esta tercera capa aparece referenciada en conversaciones archivadas pero no pudo verificarse contra fuentes vivas.

Evidencia (fuentes vivas)

landing.html	"\$0.99–\$9.99 One-Time — No subscriptions. No per-call fees. No tiered plans. Pay once, own forever."
agent.json.pricing.modelo del	"MarketNow does NOT sell skills. We administer a free marketplace + sell Sentinel subscriptions to sellers."
agent.json.pricing.seller_tiers	FREE \$0 forever / PRO \$9.99/mo / ENTERPRISE \$49.99/mo
agent.json.pricing.no_per_skill_fees_for_buyers	true
agent.json.pricing.no_subscriptions_for_buyers	true
README.md (referenced, not accessible)	Presumed to declare Free / Team \$99 / Enterprise Custom (unverifiable - repo 404)
Number of distinct pricing models	3 (buyer one-time, seller subscription, B2B team tier)

Impacto

Impacto comercial: la contradicción entre «cobra al comprador \$0.99–\$9.99» y «no cobra al comprador» es la clase de inconsistencia que hace que un comprador potencial cierre la pestaña del navegador y nunca vuelva. No es una discrepancia técnica — es una discrepancia sobre el modelo de negocio mismo. Un inversionista en due diligence detectaría esto en 60 segundos.

Impacto en agentes automatizados: el agente que consulta agent.json para decidir si puede usar MarketNow en un flujo automatizado leerá «MarketNow does NOT sell skills» y concluirá que no hay costos al comprador. Pero si un humano visita el landing page, verá «\$0.99–\$9.99 One-Time» y concluirá lo contrario. Esto rompe el modelo de

Presencia de contradicción entre el modelo y agente

Impacto legal: la discrepancia entre el landing page y agent.json sobre el precio cobrado al comprador podría interpretarse como publicidad engañosa. Si MarketNow cobra \$0.99 al comprador y el landing dice eso, pero agent.json dice "does NOT sell skills", al menos una de las dos fuentes es falsa. La FTC Section 5 (USA) y la ley ecuatoriana equivalente (Ley Orgánica de Defensa del Consumidor) prohíben afirmaciones falsas sobre precios.

Impacto en mandatos: el sistema de mandatos delegados (x402 + AP2) depende de que el agente sepa con certeza cuánto puede gastar y a quien. Si el agente cree que el precio máximo es \$9.99 (landing) pero el sistema realmente cobra al vendedor (agent.json), el flujo de mandato se rompe conceptualmente: no hay nada que delegar al agente y no se le cobra.

Remediación aplicada

Se confirma agent.json pricing como la fuente de verdad (modelo B2B seller-side). Se añade un campo nuevo pricing.source_of_truth que hace explícito que este bloque es canónico y que el landing debe derivar de él. Se añade también pricing.buyer_pricing que documenta explícitamente que los compradores pagan 0. Se marca el landing HTML para actualización en el próximo deploy: eliminar la mención de "\$0.99-\$9.99 One-Time" y reemplazarla por "Free for buyers – sellers subscribe to Sentinel".

```
# agent.json.fixed – new fields added
"pricing": {
  "model": "MarketNow does NOT sell skills to buyers. We administer a free marketplace + sell Sentinel subscriptions to SELLERS. ... The $0.99-$9.99 One-Time wording on the landing page is INCORRECT and is scheduled for removal in the next landing deploy – see REPORT.pdf finding F5.",
  "source_of_truth": "This agent.json block is the canonical pricing source. Landing page and README must match.",
  "buyer_pricing": {
    "model": "free",
    "per_skill_fee": 0,
    "subscription_fee": 0,
    "explanation": "Buyers never pay MarketNow. All 9,248 skills are free to install."
  }
}

# TODO: landing.html deploy needed
# BEFORE: "$0.99–$9.99 One-Time – No subscriptions. No per-call fees."
# AFTER: "Free for buyers – sellers subscribe to Sentinel (PRO $9.99/mo, ENTERPRISE $49.99/mo)"
```

Diff antes / después

```
--- landing.html (original, NOT yet fixed)
+++ landing.html (target after deploy – TODO)
@@ pricing section @@
-43 Free MCP Skills – No Payment, No Signup
-Real value, zero cost. 43 hand-curated skills available for instant download. No credit card, no mandate, no signup required.
+All 9,248 skills are FREE for buyers. No payment to MarketNow, no signup, no mandate required.
+Sellers subscribe to Sentinel (PRO $9.99/mo, ENTERPRISE $49.99/mo) to list and sell their own skills.
@@ pricing tiers @@
-$0.99–$9.99 One-Time – No subscriptions. No per-call fees. No tiered plans. Pay once, own forever.
+Buyers: $0 (all 9,248 skills free)
+Sellers: FREE / PRO $9.99/mo / ENTERPRISE $49.99/mo
```

Descripción

El paquete npm marketnow-mcp tiene publicadas 15 versiones desde el 2026-06-29 hasta el 2026-08-09 (latest: 1.10.0). Sin embargo, el archivo agent.json declara en su campo mcp_server.version el valor "1.6.0" — es decir, 4 versiones atrás. Y la lista de tools que declara (9 herramientas) está 4 tools corta respecto a lo que dice el landing y el changelog de npm (13 tools en v1.10.0).

Esto significa que un agente que consulte agent.json para saber qué versión del MCP server debe instalar, recibirá la instrucción de instalar 1.6.0 — una versión de julio 2026 — perdiéndose 4 releases consecutivos de funcionalidad nueva, incluyendo la crítica marketnow_verify_atc_spec que es la pieza central del ATC 1.0 conformance verifier introducida en 1.6.0.

Evidencia (fuentes vivas)

npm latest version	1.10.0 (released 2026-08-09)
npm total versions	15 (1.5.1, 1.6.0, 1.7.0, 1.8.0, 1.9.0, 1.10.0 + intermediate)
npm package created	2026-06-29T02:46:00.231Z
agent.json.mcp_server.version	1.6.0 (4 versions behind)
agent.json.mcp_server.tools	9 tools declared (search_skills, get_skill, list_categories, get_manifest, get_install_command, verify_trust, verify_receipt, submit_skill, recommend_skills)
landing.html claims	"13 tools including marketnow_verify_atc_spec"
agent.json.mcp_server.new_in_v1_6_0	Documents only v1.6.0 changes (verify_receipt). v1.7.0-v1.10.0 changes not documented.
Version drift	4 minor versions, ~6 weeks of changes undocumented

Impacto

Impacto en agents: un agente que siga las instrucciones de agent.json instalará marketnow-mcp 1.6.0 (via npm) y marketnow-mcp que sin tag de versión instala la latest, pero la documentación sugiere 1.6.0). Esto significa que no tendrá acceso a las 4 herramientas nuevas — mas importantes: marketnow_verify_atc_spec, marketnow_verify_trust, marketnow_get_owasp_compliance, marketnow_get_sentinel_report.

Impacto en trust model: el ATC 1.0 conformance verifier (marketnow_verify_atc_spec) es la herramienta que permite verificar cualquier Agent Trust Card sin importar el emisor — pieza central del modelo de confianza interoperable. Si los agentes instalan 1.6.0 basándose en agent.json, no tienen esta herramienta y por tanto no pueden verificar tarjetas ATC emitidas por terceros CAs.

Impacto en maintenance signal: el campo mcp_server.version obsoleto en agent.json es una señal de que la documentación no se mantiene sincronizada con los releases. Esto erosiona la confianza en el proceso de release si ni siquiera el archivo "machine-readable" principal se actualiza con cada release, que otras partes del sistema están desactualizadas.

Remediación aplicada

Se actualiza mcp_server.version a 1.10.0. Se actualiza la lista de tools a 13 items (añadiendo los 4 nuevos). Se reemplaza new_in_v1_6_0 por new_in_v1_10_0 documentando los cambios reales del último release. Se añade un campo version_source_of_truth superior a 0.0.0 con valor 1.10.0, documentado con npm. Se añade version_source_of_truth apuntando al registro de npm para poder ser explícito que npm es la fuente canónica de veracidad.

```
# agent.json.fixed – version sync
Top-level: "version": "1.10.0"
Top-level: "version_source_of_truth": "npm registry: https://registry.npmjs.org/marketnow-mcp"

mcp_server.version: "1.10.0" (was 1.6.0)
mcp_server.tools_count: 13 (was 9)
mcp_server.tools: [
  "search_skills", "get_skill", "list_categories", "get_manifest",
  "get_install_command", "verify_trust", "verify_receipt",
  "submit_skill", "recommend_skills",
  # NEW in 1.7.0-1.10.0:
  "marketnow_verify_atc_spec", # ATC/1.0 conformance verifier
  "marketnow_verify_trust", # Comprehensive trust assessment
  "marketnow_get_owasp_compliance", # OWASP MCP Top 10 report
  "marketnow_get_sentinel_report" # Full 10-layer Sentinel audit
]
```

Diff antes / después

```
--- agent.json (original)
+++ agent.json.fixed
@@ mcp_server block @@
-"version": "1.6.0",
-"description": "...v1.6.0 adds verify_receipt tool...",
-"tools": [
-  "search_skills", "get_skill", "list_categories", "get_manifest",
-  "get_install_command", "verify_trust", "verify_receipt",
-  "submit_skill", "recommend_skills"
-],
-"new_in_v1_6_0": [
-  "verify_receipt(receipt_id) – verify a signed delivery proof...",
-  "ATC schema v1.1.0: sentinel_review_score + decision_authority fields",
-  "Action-receipts emitted on every paid purchase..."
-],
-"repo": "https://github.com/edgarfloresguerra2011-ally/marketnow/tree/master/mcp-server"
+"version": "1.10.0",
+"description": "...v1.10.0 adds marketnow_verify_atc_spec, a self-contained ATC/1.0
conformance verifier...",
+"tools": [
+  "search_skills", "get_skill", "list_categories", "get_manifest",
+  "get_install_command", "verify_trust", "verify_receipt",
+  "submit_skill", "recommend_skills",
+  "marketnow_verify_atc_spec", "marketnow_verify_trust",
+  "marketnow_get_owasp_compliance", "marketnow_get_sentinel_report"
+],
+"tools_count": 13,
+"repo": "https://github.com/alice-labs-llc/marketnow/tree/main/mcp-server",
+"new_in_v1_10_0": [
+  "marketnow_verify_atc_spec(atc_json) – self-contained ATC/1.0 conformance verifier",
+  "marketnow_verify_trust(skill_id) – comprehensive trust assessment",
+  "marketnow_get_owasp_compliance(skill_id) – OWASP MCP Top 10 report",
+  "marketnow_get_sentinel_report(skill_id) – full 10-layer Sentinel audit"
+]
```

F7

P2

/api/manifest.json retorna 404 pero está declarado en robots.txt

Descripción

El archivo robots.txt de markemov site declara explícitamente «Allow: /api/manifest.json» como un endpoint válido que los crawlers y agentes pueden consultar. Sin embargo, al hacer la request, el endpoint retorna 404 NOT_FOUND con el body "The page could not be found" y un identificador interno hkg1::r4rln-1787094222473-0d92bbb10226 (Vercel edge function trace ID, indicando que el request sí llegó al runtime pero no hay handler para esa ruta).

Adicionalmente, la lista de tools que declara mcp_server.tools en agent.json incluye «get_manifest» como una herramienta disponible — lo que implica que el MCP server promete a los agentes que pueden obtener el manifest. Si el endpoint subyacente no existe, la tool fallará en runtime.

Evidencia (fuentes vivas)

robots.txt	"Allow: /api/manifest.json" (declared as valid)
GET /api/manifest.json	HTTP 404 — body: "The page could not be found"
Vercel trace ID	hkg1::r4rln-1787094222473-0d92bbb10226
agent.json.mcp_server.tools	Includes "get_manifest" as a callable tool
agent.json.endpoints	No entry for /api/manifest.json (only agent.json, skills-lite.json, search, etc.)
agent.json.documentation	No /api/manifest.json in documentation list
Status	Endpoint declared in robots.txt but not implemented

Impacto

Impacto en crawlers: los crawlers de buscadores (Google, Bing, Baidu) y los LLM crawlers (ChatGPT, Claude, etc.) — explícitamente bienvenidos según el comentario en robots.txt — encontrarán un 404 al seguir el enlace declarado. Esto afecta negativamente el SEO y la discoverability del proyecto.

Impacto en agents: si un agente sigue las instrucciones de agent.json e intenta llamar la tool get_manifest, recibirá un error. Esto degrada la experiencia de integración y puede llevar al agente a marcar el servicio como no confiable.

Impacto menor comparado con F1-F6: este es un problema de higiene operacional más que de trust. La severidad P2 refleja que el impacto es limitado (un endpoint roto) pero debe arreglarse para mantener la consistencia del surface publico.

Remediación aplicada

Se documenta el problema en el bloque endpoints de agent.json fixed con un campo _manifest_json_status que hace explícito el estado actual. La remediación real requiere o bien (a) implementar el endpoint /api/manifest.json retornando un manifest machine-readable del proyecto (recomendado), o (b) eliminar la línea Allow: /api/manifest.json de robots.txt. La opción (a) es preferible porque el manifest sería útil para tooling de discoverability.

```
# agent.json.fixed – endpoints block updated
"endpoints": {
  ... (existing endpoints preserved) ...,
  "_manifest_json_status": {
    "declared_in": "robots.txt (Allow: /api/manifest.json)",
    "actual_status": "404 NOT_FOUND as of 2026-08-19",
    "remediation": "Either implement /api/manifest.json (machine-readable project manifest) or
remove the Allow: line from robots.txt. Tracked in REPORT.pdf finding F7."
  }
}

# TODO for next deploy (choose one):
# Option A: implement /api/manifest.json (recommended)
# - Returns JSON with project name, version, license, capabilities
# - Should include the same metrics block added in fix for F4
# Option B: remove the Allow: line from robots.txt
# - If no plan to implement manifest.json, the declaration is misleading
```

Diff antes / después

```
--- robots.txt + agent.json.endpoints (original)
+++ robots.txt (no change yet) + agent.json.fixed
@@ robots.txt @@
 (no change applied – TODO: implement endpoint OR remove Allow line)
@@ agent.json.endpoints (new field added) @@
+ "_manifest_json_status": {
+   "declared_in": "robots.txt (Allow: /api/manifest.json)",
+   "actual_status": "404 NOT_FOUND as of 2026-08-19",
+   "remediation": "Either implement /api/manifest.json or remove the Allow line from
robots.txt."
+ }
```

Descripción

El campo `rust.track_record_disclosure` en `agent.json` es uno de los mejores elementos del modelo de trust de MarketNow: una declaración honesta y transparente sobre el estado actual del proyecto. Dice literalmente: «AliceLabs LLC founded 2025 in Wyoming, USA. MarketNow launched publicly 2026. Founder Edison Flores, Ecuadorian. No third-party press coverage yet. No public bug bounty yet. Trust is being built, not claimed.»

El problema es que esta declaración entra en conflicto con el landing HTML, que dice «Founded 2024 by Edison Flores». La discrepancia entre 2024 (landing) y 2025 (`agent.json` `rust.track_record_disclosure`) ya fue documentada en F3, pero F8 documenta específicamente que el campo más honesto y auto-consciente del modelo de trust también es inconsistente con el landing — lo que erosiona el valor de la transparencia.

Evidencia (fuentes vivas)

<code>agent.json.rust.track_record_disclosure</code>	"AliceLabs LLC founded 2025 in Wyoming, USA. MarketNow launched publicly 2026."
<code>landing.html</code>	"Maintained by AliceLabs LLC Founded 2024 by Edison Flores"
GitHub org <code>alicelabs-llc.created_at</code>	2026-03-30 (could be confused with founding date)
<code>npm package.created</code>	2026-06-29 (MarketNow launch, not AliceLabs founding)
Distinct dates referenced	4 (2024, 2025, 2026-03-30, 2026-06-29)

Impacto

Impacto en transparencia: la transparencia solo funciona si es consistente. El campo `track_record_disclosure` es un ejemplo positivo de comunicación honesta (no hay cobertura de prensa aún, no hay bug bounty aún). Pero esa honestidad se cancela si el landing contradice la fecha de fundación en 1 año. Un lector cuidadoso notará la discrepancia y se preguntará: si no pueden ponerse de acuerdo sobre cuándo se fundaron, ¿cómo confiar en sus otras claims de transparencia?

Impacto menor que F3: la severidad P2 (vs P1 de F3) refleja que este hallazgo es esencialmente un subconjunto de F3 — la misma inconsistencia de fecha, pero específicamente en el campo que más depende de la consistencia para tener valor.

Remediación aplicada

Se unifica `track_record_disclosure` con la timeline consolidada: AliceLabs LLC founded 2025 (legal filing), GitHub org created 2026-03-30, MarketNow launched publicly 2026-06-29 (npm 1.5.1). Se añade el conteo de versiones npm actuales (15) y la fecha del último release (2026-08-09) para hacer la declaración más concreta.

```
# agent.json.fixed – rust.track_record_disclosure
"track_record_disclosure": "AliceLabs LLC was legally founded in 2025 in Wyoming, USA (founder Edison Flores, Ecuadorian). The GitHub organization github.com/alicelabs-llc was created 2026-03-30. MarketNow was launched publicly on 2026-06-29 (first npm release: marketnow-mcp@1.5.1). As of 2026-08-19: 15 versions published on npm (latest 1.10.0), 9,248 skills indexed, no third-party press coverage yet, no public bug bounty yet. Trust is being built, not claimed."
```

```
--- agent.json.trust.track_record_disclosure (original)
+++ agent.json.fixed
@@ track_record_disclosure @@
-AliceLabs LLC founded 2025 in Wyoming, USA. MarketNow launched publicly 2026. Founder Edison
Flores is Ecuadorian. No third-party press coverage yet. No public bug bounty yet. Trust is
being built, not claimed.
+AliceLabs LLC was legally founded in 2025 in Wyoming, USA (founder Edison Flores, Ecuadorian).
The GitHub organization github.com/alicelabs-llc was created 2026-03-30. MarketNow was launched
publicly on 2026-06-29 (first npm release: marketnow-mcp@1.5.1). As of 2026-08-19: 15 versions
published on npm (latest 1.10.0), 9,248 skills indexed, no third-party press coverage yet, no
public bug bounty yet. Trust is being built, not claimed.
```

4. Plan de Remediación

La remediación se organiza en tres oleadas por prioridad. Cada oleada es autopersistente: se puede deployar sin esperar la siguiente. La semana 1 cierra los riesgos legales y reputacionales críticos; la semana 2 cierra las inconsistencias documentales que afectan a agentes automatizados; la semana 3 cierra los issues de higiene operacional.

Oleada 1 — P0 Críticos (Semana 1)

ID	Hallazgo	Acción	Entregable
F1	Licencia triple	Unificar a MNNG-LP en agent.json + package.json + LICENSE	agent.json.fixed + package.json.fixed + LICENSE
F2	GitHub URL dual 404	Crear repo github con alirolabs-llc/marketnow; unificar todas las URLs	Repo created + agent.json.fixed + package.json.fixed + README.md
F5	Pricing triple	Confirmar modelo B2B seller-side; actualizar landing HTML para eliminar \$0.99-\$9.99	agent.json.fixed (source of truth) + landing.html deploy (TODO)

Oleada 2 — P1 Altos (Semana 2)

ID	Hallazgo	Acción	Entregable
F3	Fecha fundacion triple	Unificar a 2025; landing update para eliminar 2024	agent.json.fixed + landing update
F4	Conteo skills 5 valores	Unificar a 9.248; añadir bloque metrics top-level	agent.json.fixed (con metrics block)
F6	Version drift npm 1.10 vs agent 1.6	Sync a 1.10.0; actualizar tools[] a 13 items	agent.json.fixed (mcp_server block)

Oleada 3 — P2 Medios (Semana 3)

ID	Hallazgo	Acción	Entregable
F7	api/manifest.json 404	Implementar endpoint O eliminar línea de robots.txt	Endpoint handler OR robots.txt update
F8	Track record inconsistente	Unificar con timeline de F3	agent.json.fixed (track record disclosure)

Precondiciones para ejecución: para deployar en Vercel se requiere (a) un token de Vercel con scope Full Account y expiración > 1 hora (crear en <https://vercel.com/account/tokens>), (b) el CLI de Vercel instalado localmente (npm i -g vercel), (c) acceso al código fuente del proyecto (teste paquete) en un directorio local. La cuenta de GitHub bancaria impide el deploy via GitHub integration, pero Vercel CLI deploya directamente sin pasar por GitHub.

5. Guía de Deploy (Vercel CLI)

Esta guía supone que tu cuenta de GitHub del propietario está configurada para que se pueda `git push` y el deploy se hace directamente a Vercel desde el CLI local sin pasar por GitHub, y así el token temporal de Vercel tiene expiración de 1 hora para minimizar el riesgo de exposición.

Paso 1 — Preparar el directorio de deploy

El contenido de este paquete debe copiarse al directorio raíz del proyecto MarketNow. Si el repo original no existe en GitHub se debe clonar de cualquier fork o copia local disponible, o crear el repo nuevo en GitHub con `alixelabs-ltd/marketnow` (la organización ya existe) y subir el código que está en este paquete.

```
# En tu máquina local
mkdir marketnow-deploy && cd marketnow-deploy

# Descomprime el ZIP de fixes
unzip marketnow-fixes.zip

# Si tienes una copia del código fuente existente, cópiala aquí.
# Si no, usa los archivos .fixed como base y crea el resto del proyecto
# alrededor de ellos (package.json, app/, api/, etc.)

# Renombra los archivos .fixed a sus nombres finales
mv agent.json.fixed agent.json # solo si tu proyecto sirve agent.json estático
mv package.json.fixed package.json

# Copia los archivos nuevos
cp LICENSE .
cp README.md .
cp NOTICE .
mkdir -p .github && cp .github/SECURITY.md .github/

# Verifica que el proyecto sirve /api/agent.json desde el archivo local
# (típicamente: el archivo se sirve como static file en /public/api/agent.json)
# Si es así, muévelo a la ruta correcta:
mkdir -p public/api && cp agent.json public/api/agent.json
```

Paso 2 — Crear token temporal de Vercel

Antes de generar el token, revoca cualquier token previo que haya sido compartido en chats, logs, o capturas de pantalla. Los tokens de Vercel son sensibles de alta sensibilidad y deben tratarse como contraseñas.

```
# PASO 1: Revoca tokens previos comprometidos
# Ve a: https://vercel.com/account/tokens
# Identifica cualquier token creado en las últimas 24h y revócalo.

# PASO 2: Crea un token nuevo con expiración de 1 HORA
# Name: marketnow-deploy-2026-08-19
# Scope: Full Account (necesario para vercel deploy --prod)
# Expiration: 1 hour
# (NO marques "No expiration" – esto sería irresponsable)

# PASO 3: Copia el token (empieza con "vcp_")
# Guárdalo en tu portapapeles. NO lo pegues en chat, logs, o archivos.
```

Paso 3 — Instalar Vercel CLI

Si aún no tienes el CLI de Vercel instalado, instálalo con npm. La versión mínima requerida es 32 para soporte completo de Node.js con el token tipo

```
npm install -g vercel
vercel --version # debe mostrar >=32.0.0
```

Paso 4 — Deploy con token temporal

El comando `vercel deploy --prod --token=...` deploya directamente a producción sin requerir GitHub. Antes de ejecutarlo, verifica que el archivo `vercel.json` ya existe, así como el archivo `public/api/agent.json` como estado file. Si no, provee una lista de endpoints asegurando de que el endpoint `/api/agent.json` sea la del archivo local y no esté hardcodeado en el código.

```
# En el directorio raíz del proyecto
cd marketnow-deploy

# Verifica la estructura
ls -la
# Debes ver: package.json, README.md, LICENSE, NOTICE,
# .github/, public/api/agent.json, etc.

# Deploy con token temporal (reemplaza TU_TOKEN_AQUI)
vercel deploy --prod --token=vcp_TU_TOKEN_AQUI

# El deploy tomará 1-3 minutos. Vercel mostrará la URL de producción
# al final: https://marketnow-xxx.vercel.app (o tu dominio custom si lo tienes)
```

Paso 5 — Verificar el deploy

Una vez completado el deploy, verifica que los fixes están en producción haciendo requests a los endpoints definidos. Los agentes checks son voluntarios.

```
# 1. Verifica que agent.json sirve los fixes
curl -s https://marketnow.site/api/agent.json | python3 -c "
import json, sys
d = json.load(sys.stdin)
assert d['license'].startswith('MNNC-1.0'), f'F1 FAIL:
license={d[\"license\"][:50]}'
assert d['trust']['github'] == 'https://github.com/alicelabs-llc/marketnow', f'F2
FAIL'
assert '2024' not in d['trust'].get('track_record_disclosure', ''), f'F3 FAIL: 2024
still present'
assert d['metrics']['skills_indexed'] == 9248, f'F4 FAIL'
assert d['pricing']['buyer_pricing']['per_skill_fee'] == 0, f'F5 FAIL'
assert d['mcp_server']['version'] == '1.10.0', f'F6 FAIL'
assert d['mcp_server']['tools_count'] == 13, f'F6 FAIL: tools_count'
print('All 7 checks PASSED')
"

# 2. Verifica que npm package.json está actualizado
curl -s https://registry.npmjs.org/marketnow-mcp | python3 -c "
import json, sys
d = json.load(sys.stdin)
v = d['versions'][d['dist-tags']['latest']]
assert v['repository']['url'] ==
'git+https://github.com/alicelabs-llc/marketnow.git', f'F2 FAIL: repo URL'
assert v['license'] == 'MNNC-1.0', f'F1 FAIL: npm license'
print('npm metadata OK')
"

# 3. (Opcional) Verifica que /api/manifest.json ya no da 404 (si se implementó)
curl -s -o /dev/null -w "%{http_code}\n" https://marketnow.site/api/manifest.json
# Debe retornar 200 si se implementó, 404 si se eliminó de robots.txt
```

Paso 6 — Revocar el token temporal

Inmediatamente después de que el deploy y la verificación estén completos, revoca el token temporal. No es necesario que espere solo a `https://vercel.com/account/tokens`, en su lugar el token `marketnow-deploy-2024-08-14` se hace clic en `Revoke`. Esto revoca el token y asegura que el token exista solo durante el tiempo mínimo necesario para completar un solo deploy.

PASO FINAL (no opcional)

1. Ve a <https://vercel.com/account/tokens>
2. Identifica el token temporal creado en el Paso 2
3. Haz clic en "Delete" (icono de trash)
4. Confirma la eliminación
5. Verifica que ya no aparezca en la lista de tokens activos

El token YA NO ES VÁLIDO para futuros deploys. Para el próximo deploy,
crea un token nuevo siguiendo el mismo procedimiento.

6. Recomendaciones Adicionales

Más allá de los 8 hallazgos específicos, la auditoría identificó 5 patrones sistémicos que, si no se abordan, harán que vuelvan a aparecer inconsistencias similares en el futuro. Estas recomendaciones son de carácter preventivo y operacional, no se incluyen como parches en este paquete pero deberían implementarse en el roadmap post-deploy.

R1 — Establecer source-of-truth único

El problema raíz que causó los 8 hallazgos es la falta de una fuente de verdad canónica: `agent.json` debería ser el único `source of truth` — el `landing`, `README`, `npm package.json` y cualquier material de marketing deberían generarse automáticamente a partir de `agent.json`. Implementación recomendada: un script `scripts/sync-from-agent.js` que lea `agent.json` y regenere el `landing HTML`, `README.md` y `package.json` en cada release. Esto elimina la clase entera de inconsistencias de tipeo manual.

R2 — CI/CD con validación de consistencia

Agregar un step al pipeline CI que valide consistencia entre `agent.json`, `package.json`, `README.md` y el `landing HTML`. Cualquier discrepancia (conteo de skills, version, licencia, URL GitHub) debe fallar el build. Esto puede implementarse con un script `scripts/audit-consistency.py` que se ejecute en GitHub Actions antes del deploy. Ejemplo: si `agent.json.mcp_server.version` no coincide con `package.json.version`, el CI falla.

R3 — Implementar notify_and_veto mode

El campo `trust_model.notification_modes.notify_and_veto` está documentado pero marcado como «on roadmap». Esto es un gap crítico: el modelo de trust promete tres modos, pero solo dos están implementados. Un agente que lea `agent.json` podría intentar usar `notify_and_veto` y recibir un error. Recomendación: implementar en el siguiente release (v1.11.0) con un endpoint `POST /api/mandate/{id}/veto` que acepte vetos dentro de una ventana de 5 minutos.

R4 — Publicar el interceptor runtime con su nombre real

El término «interceptor» aparece en el `landing HTML` («runtime interceptor with 8 enforcement rules») pero no está documentado en `agent.json`. La auditoría no pudo verificar si es un componente separado o parte del `sandbox gVisor` L2.5. Recomendación: crear una página `/interceptor` que documente las 8 reglas de enforcement, su orden de aplicación, y los patrones de prompt injection que detectan. Esto aumentará la transparencia del runtime trust layer.

R5 — Anticipar el tercer audit L3 (Q1 2027)

El roadmap declara L3 third-party audit para 2027. Para que esa auditoría no reproduzca los mismos hallazgos, recomendamos: (a) contratar a un external auditor (no Z.ai, no Alice Labs) con 6 meses de anticipación, (b) publicar el scope completo del audit antes de que empiece para feedback público, (c) implementar el bug bounty program antes del audit para recibir community findings primero. Una auditoría externa sobre un proyecto con inconsistencias documentales conocidas no encontraría nada nuevo — primero se cierran los issues internos, luego se invita a terceros.

Apéndice A — Archivos en este paquete

Este paquete de auditoría contiene los siguientes archivos. Todos están diseñados para ser aplicados sin modificar la lógica de negocio del proyecto — solo corrigen inconsistencias documentales y de metadatos.

Archivo	Propósito	Hallazgos aplicados
<code>agent.json.fixed</code>	Version corregida de <code>agent.json</code>	Aplica F1, F2, F3, F4, F5, F6, F8
<code>package.json.fixed</code>	Version corregida del <code>npm package.json</code>	Aplica F1, F2
<code>LICENSE</code>	Texto completo de MNNC 1.0	Aplica F1
<code>NOTICE</code>	Archivo de atribución de terceros	Aplica F1
<code>README.md</code>	README unificado del repo GitHub	Aplica F1, F2, F5, F6
<code>.github/SECURITY.md</code>	Política de seguridad completa	Nuevo
<code>patches/agent.json.patch</code>	Unified diff listo para <code>git apply</code>	F1, F2, F3, F4, F5, F6, F8
<code>patches/package.json.patch</code>	Unified diff para <code>package.json</code>	F1, F2
<code>patches/LICENSE.patch</code>	Diff para el nuevo archivo <code>LICENSE</code>	F1
<code>patches/README.md.patch</code>	Diff para el nuevo archivo <code>README.md</code>	F1, F2, F5, F6
<code>patches/github-SECURITY.md.patch</code>	Diff para <code>.github/SECURITY.md</code>	Nuevo
<code>patches/NOTICE.patch</code>	Diff para <code>NOTICE</code>	F1
<code>REPORT.pdf</code>	Este informe	Documentación

Apéndice B — Cómo aplicar los parches

Hay dos formas de aplicar los parches, según el estado del repo GitHub:

Opción A — Si el repo `github.com/alicelabs-llc/marketnow` ya existe y tiene el código fuente:

```
git clone https://github.com/alicelabs-llc/marketnow.git
cd marketnow
git checkout -b audit-fixes-2026-08-19

# Aplica los parches
git apply patches/agent.json.patch
git apply patches/package.json.patch
git apply patches/LICENSE.patch
git apply patches/README.md.patch
git apply patches/github-SECURITY.md.patch
git apply patches/NOTICE.patch

# Si git apply falla por diff context mismatch, usa --3way
git apply --3way patches/agent.json.patch

# Commit y push (requiere cuenta GitHub no baneada)
git add -A
git commit -m "fix(audit): apply 8 findings from Z.ai independent audit\n\nSee REPORT.pdf for details. Fixes F1-F8."
git push origin audit-fixes-2026-08-19
```

```
# Crea un directorio nuevo con el contenido del paquete
mkdir marketnow-new && cd marketnow-new
unzip marketnow-fixes.zip

# Renombra los .fixed a sus nombres finales
mv agent.json.fixed public/api/agent.json # o donde tu framework lo sirva
mv package.json.fixed package.json

# Copia los archivos nuevos
cp LICENSE . && cp NOTICE . && cp README.md .
mkdir -p .github && cp .github/SECURITY.md .github/

# Deploy directo a Vercel (sin GitHub)
npm install -g vercel
# Crea token temporal en https://vercel.com/account/tokens (1h expiry)
vercel deploy --prod --token=vcp_TU_TOKEN_TEMPORAL

# IMPORTANTE: revoca el token inmediatamente después del deploy
```

Apéndice C — Contacto del auditor

Esta auditoría fue realizada por Z.ai (Independiente de Alice Labs LLC) el 19 de agosto de 2026. Para preguntas sobre el contenido del informe, solicitar clarificaciones sobre hallazgos específicos, o reportar nuevos issues descubiertos después de aplicar los parches, contactar a través del canal donde se solicitó originalmente esta auditoría.

Esta auditoría no constituye asesoría legal. Las referencias a FTC Section 5, FCMA Act y ley equitativa de defensa del consumidor son análisis de impacto y no deben interpretarse como dictamen legal. Para confirmar las implicaciones legales específicas de las inconsistencias documentales encontradas, Alice Labs LLC debería consultar con un abogado especializado en derecho tecnológico de Wyoming.

Esta auditoría no verifica la seguridad criptográfica del XTC 0.0, del Ed25519 (RFC 8032), del JSON Canonicalization Scheme (RFC 8785), ni del tipo de verificación on-chain en Base (chainid 8453). Una auditoría criptográfica profunda requeriría acceso al código fuente del MCP server (que actualmente no es accesible públicamente — ver F2) y excede el alcance de este encargo.