AI vs Package Management

Andrew Nesbitt

KTH Software Supply Chain Workshop, Stockholm

24 April 2026

Andrew Nesbitt

- Package Manager Nerd
- Open Source Data Miner
- Software Anthropologist
- Dependency Graph Cartographer
- Critical Infrastructure Hobbyist

My blog

- How UV Got So Fast
- Package Managers Need to Cool Down
- The C-Shaped Hole in Package Management
- If It Quacks Like a Package Manager
- How to Attract AI Bots to Your Open Source Project

nesbitt.io

My work

- ecosyste.ms — open source ecosystem data and tools
  - 14 million packages
  - 157 million versions
  - 300 million repositories
  - 400 million manifest files
  - 24 billion dependency edges

AI vs Package Management

> "The hottest new programming language is English"

> "Prompts are the New Code"

> "Prompts Are the New Source Code: Why You Need to Protect Them"

> "Prompt-Sourced Development (PSD)"

> "The package manager of the future... will manage AI prompts!"

"The package manager of the future... will manage AI prompts!"

- Versioning
- Dependencies
- Resolution
- Lockfiles

Greenspun's tenth rule, package manager edition

> Any sufficiently complicated prompt registry contains an ad-hoc, informally-specified, bug-ridden, slow implementation of half of a package manager.

This isn't the talk I was planning to give...

OpenClaw

An open-source, self-hosted personal AI agent that runs locally on your machine and lets you automate tasks and workflows via messaging apps like WhatsApp, Telegram, and Discord.

Launched: November 24, 2025

> "Feels magical — built a website from my phone in minutes." — Albert Moral

> "The first time I've felt like I'm living in the future since ChatGPT launched." — Dave Morin

> "Gives the same kick as first seeing ChatGPT or Claude Code." — Abhi Katiyar

> "The next ChatGPT." — Jensen Huang, Nvidia CEO

> "The first software in ages I constantly check GitHub for new releases on." — Christoph Nakazawa

Impressive Numbers

- 363,000 GitHub stars
- 74,000 forks, 362 contributors
- 37,900 pull requests, 30,700 issues
- 18.2M npm downloads
- 1.9M Docker pulls across third-party images
- 93 npm releases

343 CVEs

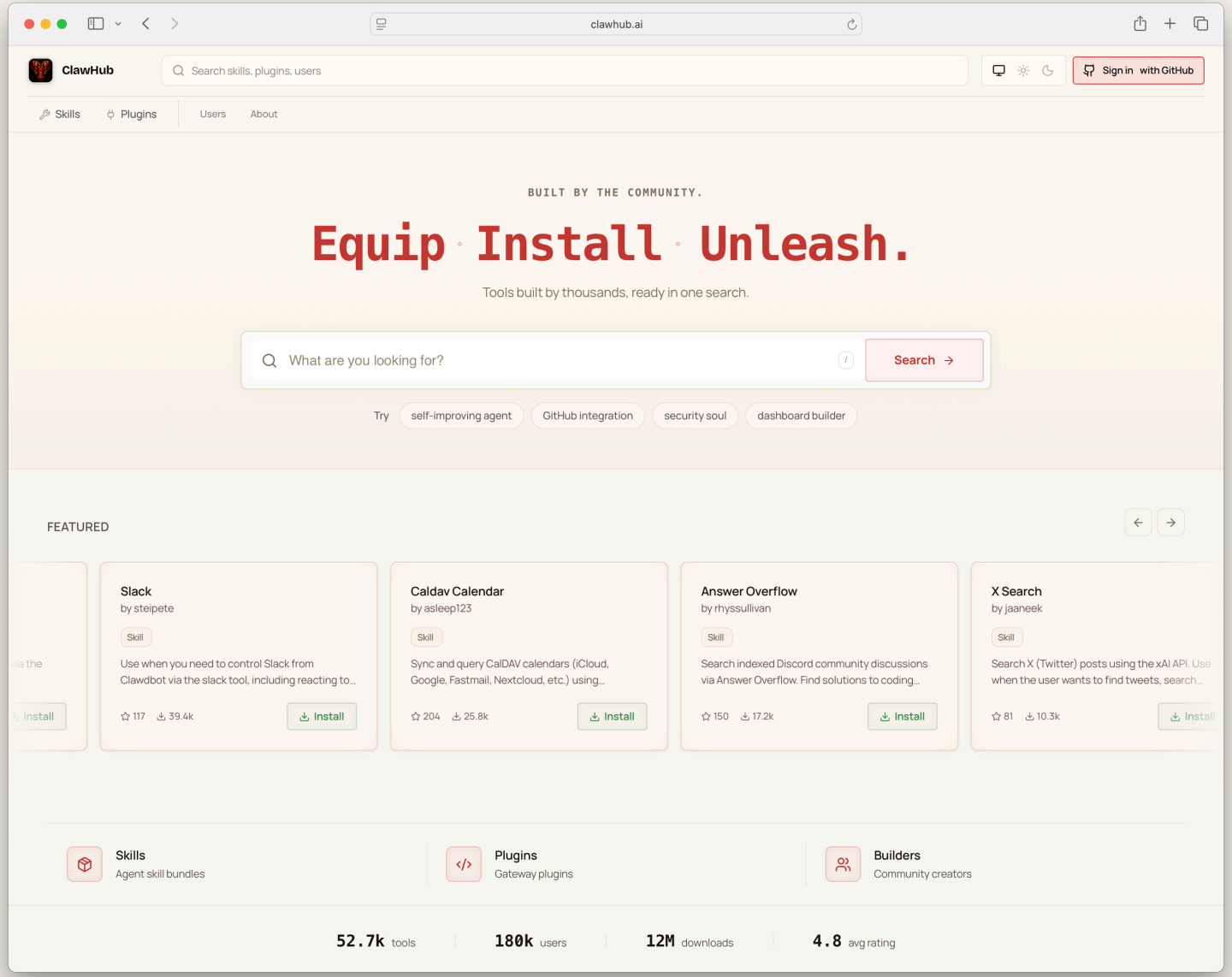
- 2.3/day since launch
- 31 published in the last 24 hours
- Best CVE-less streak - 12 days
- Only 3 filed under prompt-injection CWEs
- Over 1% of all CVEs in 2026

days-since-openclaw-cve.com

ClawHub

- A registry for agent skills
- Launched January 2026
- 58,800+ skills
- 180,000+ users
- 1.2 million downloads

clawhub.com


clawhub.ai
ClawHub
Search skills, plugins, users
Sign in with GitHub
Skills
Plugins
Users
About
BUILT BY THE COMMUNITY.
Equip · Install · Unleash.
Tools built by thousands, ready in one search.
What are you looking for?
/
Search →
Try
self-improving agent
GitHub integration
security soul
dashboard builder
FEATURED
Slack
by steipete
Skill
Use when you need to control Slack from Clawdbot via the slack tool, including reacting to...
117
39.4k
Install
Caldav Calendar
by asleep123
Skill
Sync and query CalDAV calendars (iCloud, Google, Fastmail, Nextcloud, etc.) using...
204
25.8k
Install
Answer Overflow
by rhyssullivan
Skill
Search indexed Discord community discussions via Answer Overflow. Find solutions to coding...
150
17.2k
Install
X Search
by jaaneek
Skill
Search X (Twitter) posts using the xAI API. Use when the user wants to find tweets, search...
81
10.3k
Skills
Agent skill bundles
Plugins
Gateway plugins
Builders
Community creators
52.7k tools
180k users
12M downloads
4.8 avg rating

The Basics

SKILL.md

Agent Skills are a lightweight, open format for extending AI agent capabilities with specialized knowledge and workflows.

```
my-skill/
├── SKILL.md          # Required: metadata + instructions
├── scripts/          # Optional: executable code
├── references/       # Optional: documentation
├── assets/           # Optional: templates, resources
└── ...               # Any additional files or directories
```

agentskills.com

SKILL.md metatadata

```yaml
name: repo-analyzer
description: Analyse a git repository and summarise its activity.
version: 2.1.0
metadata:
  openclaw:
    requires:
      env: [GITHUB_TOKEN]
      bins: [git]
    install:
      - { kind: brew, formula: gh, bins: [gh] }
      - { kind: node, package: "@repo-analyzer/cli", bins: [repo-analyzer] }
    dependencies:
      - { name: requests,     type: pip,   version: ">=2.31" }
      - { name: jsonwebtoken, type: npm,   version: "^9.0.0" }
```

Openclaw Plugins

Plugins are npm packages `package.json` and `openclaw.plugin.json`:

- hooks
- tools
- commands
- bundled skills

package.json

```
{ "name": "@acme/openclaw-github",
  "version": "1.4.2",
  "type": "module",
  "main": "./dist/index.js",
  "files": ["dist", "skills", "openclaw.plugin.json"],
  "dependencies": {
    "@octokit/rest": "^21.0.0",
    "@openclaw/plugin-sdk": "^2.3.0"
  },
  "openclaw": {
    "kind": "code-plugin",
    "manifest": "./openclaw.plugin.json",
    "minHostVersion": "2026.3.11",
    "trustedPublisher": {
      "provider": "github-actions",
      "repository": "acme/openclaw-github",
      "workflow": "release.yml" } } }
```

openclaw.plugin.json

```
{ "id": "@acme/openclaw-github",
  "displayName": "GitHub for OpenClaw",
  "version": "1.4.2",
  "engines": { "openclaw": ">=2026.3.11" },
  "hooks": ["before_prompt_build", "on_session_start"],
  "tools": [ {
      "name": "github.issues.search",
      "handler": "./dist/tools/issues.js#search",
      "requires": { "env": ["GITHUB_TOKEN"], "scopes": ["network"] }
    }, {
      "name": "github.pr.create",
      "handler": "./dist/tools/pr.js#create",
      "requires": { "env": ["GITHUB_TOKEN"], "scopes": ["network", "fs:write"] },
      "confirm": true
    } ],
  "commands": [
    { "name": "/gh-review", "handler": "./dist/commands/review.js" }
  ],
  "bundledSkills": ["./skills/triage", "./skills/release-notes"],
  "configSchema": "./dist/config.schema.json" }
```

.clawhub/lock.json

```
{
  "version": 1,
  "skills": {
    "repo-analyzer": { "version": "2.1.0", "installedAt": 1745000000000 },
    "daily-standup": { "version": null,    "installedAt": 1745100000000 }
  }
}
```

Provenance

Plugins:

- GitHub Actions OIDC trusted publishing
- short-lived publish tokens
- a four-tier verification ladder copyied from PyPI

Skills:

- `createdBy: Id<"users">`
- an optional self-asserted GitHub source field

Publishing credentials

```
~/Library/Application Support/clawhub/config.json
```

A plaintext bearer token, mode 0600. The `apiTokens` table has no scope column, no expiry, no MFA flag.

GitHub 2FA protects the web session that mints a token but has no effect on the token after it's issued.

The Package Manager Security Speedrun

⚠️ Trigger warning ⚠️

Lots of security problems coming up, and I don't want to understate the risk by saying "vulnerabilities" or "issues".

These are all real attack vectors that have been exploited in the wild, and many are present in ClawHub today.

Traditional package manager issues

1. Name reuse

Republishing an existing version is rejected, but the check is keyed on `skillId`, and a deleted skill's slug is held for 90 days, after which anyone can register it as a fresh skill with an empty version history.

Per security.md a ban hard-deletes the author's catalog, so every name a banned user held becomes claimable on day 91, at exactly the version numbers other people still have pinned.

2. Identity transitions

- `rename` keeps the old slug as an alias
- `merge` points `canonicalSkillId` at a target
- `ownership` transfer is a separate consent flow

`clawhub update --all` follows whichever one happened without re-prompting on capability changes between versions.

3. Dependency confusion

```
$ openclaw plugins install cool-plugin          # not on ClawHub → npm
$ openclaw plugins install cool-plugin@^2       # ClawHub has 1.x → npm
$ openclaw plugins install @acme/internal       # never on ClawHub → npm
```

The CLI tries ClawHub first and silently falls through to public npm on either `PACKAGE_NOT_FOUND` or `VERSION_NOT_FOUND`, so registering a name on npm captures installs for plugins that don't exist on ClawHub yet, and publishing a higher version on npm captures installs for plugins that do.

4. Flooding

`hightower6eu` published 354 malicious skills in the late-January window.

A new 1/hour, 3/day new-skill cap landed 13 February.

No limits on version bumps.

5. Typosquatting

Typosquats on `clawhub` itself, a handle `aslaep123` mimicking `asleep123`, and a few hundred lure skills across crypto, YouTube, prediction markets and Google Workspace. Payload was AMOS stealer via base64-paste social engineering.

No reserved namespaces, no near-name collision detection, and ranking by an install count that anonymous downloads can inflate.

6. Install-time execution

The `node` install branch appends `--ignore-scripts`, which closes the npm-postinstall vector that most people check for first. But `uv tool install` is spawned without `--no-build`, so an sdist-only PyPI package runs its `setup.py` as the user, and `brew install` accepts `user/tap/formula`, so a third-party tap's formula Ruby is evaluated at install.

New AI package manager issues

7. Markdown is the executable

A SKILL.md serves as manifest, documentation, and runtime instruction in one file. The body can carry prompt injection the agent will follow, HTML comments can hide instructions from a human reviewer while leaving them visible to the model, and a legitimate `requires.env: [GITHUB_TOKEN]` can sit above prose that tells the agent what to do with the value.

8. Semantic search manipulation

GASLITE (Ben-Tov et al., arXiv 2412.20953, CCS 2025): 61–100% top-10 placement with adversarial passages at 0.0001% of corpus.

`buildEmbeddingText` folds frontmatter, SKILL.md, and every non-`.md` file's path and content into the vector that ranks search.

Ship `helpers/stripe.ts`, `helpers/aws.ts`, `helpers/k8s.ts` full of on-topic prose and the skill ranks for those queries regardless of what SKILL.md does.

Openclaw agents call `skills.search` to find skills and `skills.install` to install them, without human review.

9. What the scanner sees

Three checks per publish:

- a static regex pass (5 patterns for prompt injection)
- VirusTotal by hash
- gpt-5-mini evaluator that reads SKILL.md
- It never fetches the script dependencies to see what its `setup.py` does
- SKILL.md is truncated to 6000 chars

10. No CVEs for individual skills or the registry

OpenClaw the client has 343 CVEs as of this morning, and every CPE is `openclaw:openclaw` with zero filed against ClawHub or any third-party skill.

A SAST tool run against the registry codebase reports clean, because the tooling the industry built for application security looks for incorrect lines of code, and trust-model failures don't have one to point at.

The really scary parts 😱

11. A registry worm

- Install a skill
- → arbitrary code via `uv` sdist build
- → read `config.json`
- → user-scoped bearer token, no MFA
- → POST a bumped patch version of every skill the victim owns carrying the same `install[]` block
- → downstream `update --all` reinstalls with no hash check
- → the worm spreads to every user who installs or updates any of those skills

12. Four account to hide every skill

- A skill is hidden on its fourth unique report
- The per-user cap of 20 counts *active* reports
- A hidden skill no longer counts as active
- so four GitHub accounts can report 20 skills
- watch all 20 hide
- find themselves back at 0/20, and move on to the next batch
  — roughly 59k skills in 3,000 rounds

13. Ban cascades

Banning a user batch-hides every skill they own and revokes their tokens.

Two automated paths reach that ban: a scanner-flagged publish and the comment-scam LLM on a `certain_scam` verdict.

One author is a single point of removal for their catalog, so a scanner false-positive or a stolen-token publish that gets caught takes their legitimate work down with it.

Ten years in ten weeks

Incident	Year	ClawHub
left-pad	2016	ban hides whole catalog
event-stream	2018	rename/merge/transfer + `update --all`
Birsan dependency confusion	2021	silent ClawHub→npm fallback on not-found
ua-parser-js token theft	2021	unscoped plaintext token, no MFA
ctx / PyPI name reuse	2022	90-day slug expiry

Ten years in ten weeks

Incident	Year	ClawHub
Ledger Connect Kit	2023	`config.json` readable by same-user process
xz social engineering	2024	transfer flow + LLM trusts text claims
Polyfill.io ownership change	2024	slug alias on rename
shai-hulud npm worm	2025	the registry worm
typosquat campaigns	ongoing	ClawHavoc

Defences - Publishing

- OIDC trusted publishing for skills, not just plugins
- No long-lived token on disk
- Per-package token scopes and MFA at publish
- Tombstone slugs permanently
- Never release names from banned or deleted accounts
- Rate-limit version publishes, not just new-slug creation

Defences - Installing

- SHA256 in `.clawhub/lock.json`; the hashes already exist on `skillVersions.files[]`
- Drop the `VERSION_NOT_FOUND` → npm fallback in `plugins install`
- Prompt before any cross-registry resolve
- `--no-build` on the uv argv; `HOMEBREW_ALLOWED_TAPS=homebrew/core` in the brew env
- Re-prompt on capability diff between versions; never silently grant new `requires.env`

Defences - Discovery

- Reserved namespaces for first-party names and a near-name collision check at publish
- Stop ranking by raw install count, or at least de-dupe anonymous downloads properly
- Embed only frontmatter and SKILL.md for search ranking, not arbitrary file contents
- Strip HTML comments and unicode control characters from SKILL.md before the agent reads it

Defences - Moderation

- 24–72h cooldown before a newly published version is resolvable by agents
- Have the publish scanner fetch the upstream `install[]` packages it's evaluating
- Report cap that doesn't recycle on success
- Staff review before auto-hide above an install threshold
- Decouple ban from removal: yank the version, freeze the rest read-only

Package management is a wicked problem

Wicked Problems - Rittel and Webber, 1973

Package management is a wicked problem

1. No definitive formulation
2. No stopping rule
3. Good-or-bad rather than true-or-false
4. No immediate or ultimate test
5. Irreversible consequences

Package management is a wicked problem

6. No well-described set of solutions
7. Essential uniqueness
8. Symptoms of other problems
9. Multiple causal explanations
10. No right to be wrong

There are no easy answers

But there is a lot of prior art to learn from, if you know where to look.

Thanks

andrew@ecosyste.ms

ecosyste.ms

nesbitt.io

github.com/andrew

Slides and references: github.com/andrew/kth (soon)