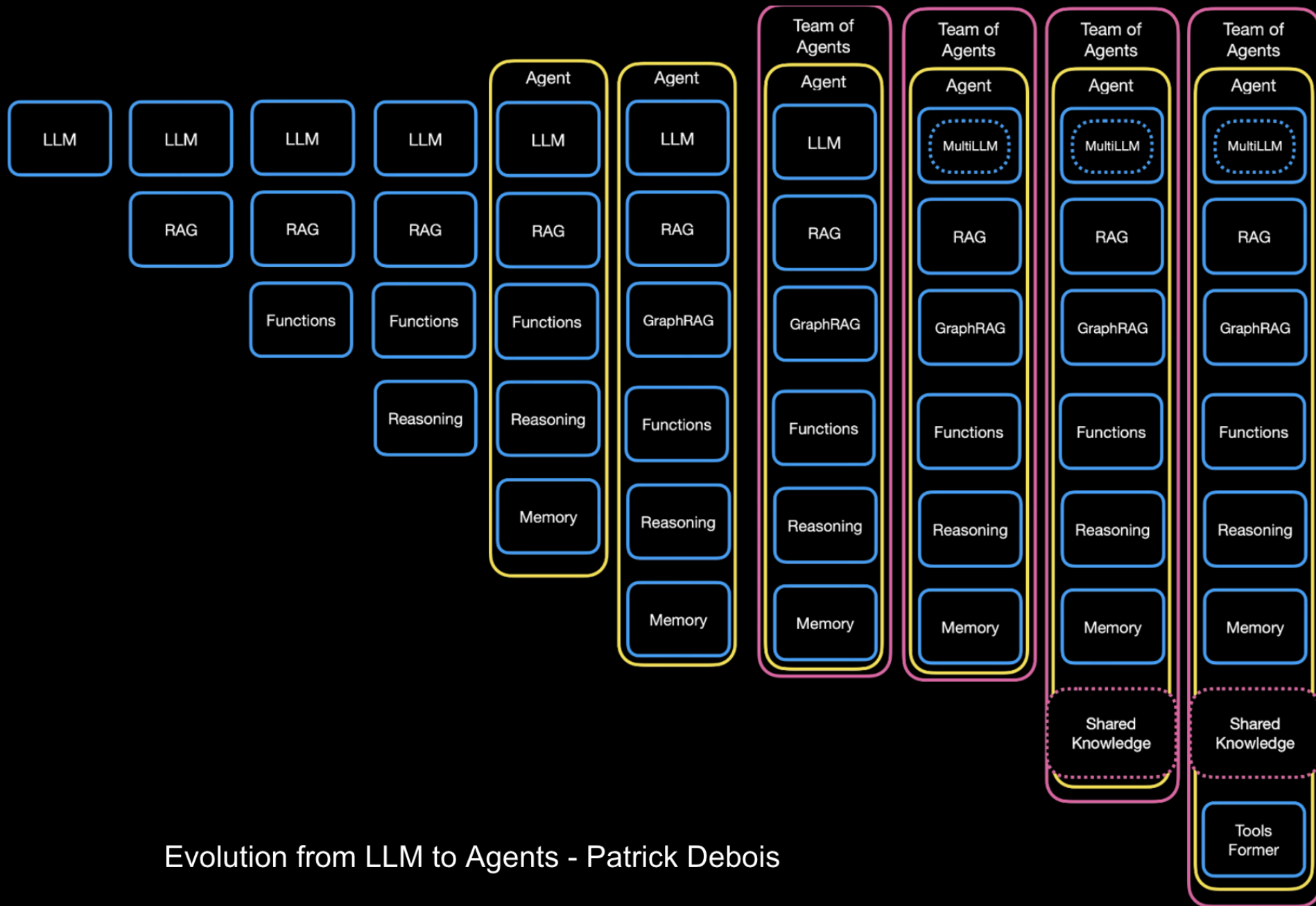


# AI Coding Fabric

Context Development Lifecycle

Agentics Day 2026 - Patrick Debois





Evolution from LLM to Agents - Patrick Debois

# Context is the new Code



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

11:17 PM · Feb 2, 2025 · **6.8M** Views



Relevant ▾

View quotes >



Post your reply

Reply

Vibe coding

This also becomes the delivery mechanism for the self-serve POC conversation. Someone's friend sends them a URL or says "run `tessl init`" — they do it, open their editor, and their agent already knows what Tessel is and what to do with it. No guided setup required.

an example of AGENTS file:

```
# Tessel

Tessel is a skill management platform for AI coding assistants. Skills are
context files that guide you on how to work within this codebase -
conventions, patterns, architecture decisions, and task-specific
instructions.

## How to work with this project

Before starting any task, check if a relevant skill exists. Skills contain
the accumulated knowledge of how work gets done here - following them
produces better, more consistent results.

**Always:**
1. Run `tessel_list_skills` if you haven't oriented yourself yet
2. Fetch any skill relevant to the task with `tessel_get_skill` before
starting
3. Follow the skill's instructions throughout the task
4. If no skill exists for what you're doing, note it - a skill should
```

Product onboarding as a skill

# I to think in parallels

2009 - What if Ops was like Dev

2025 - What if Context was like Code

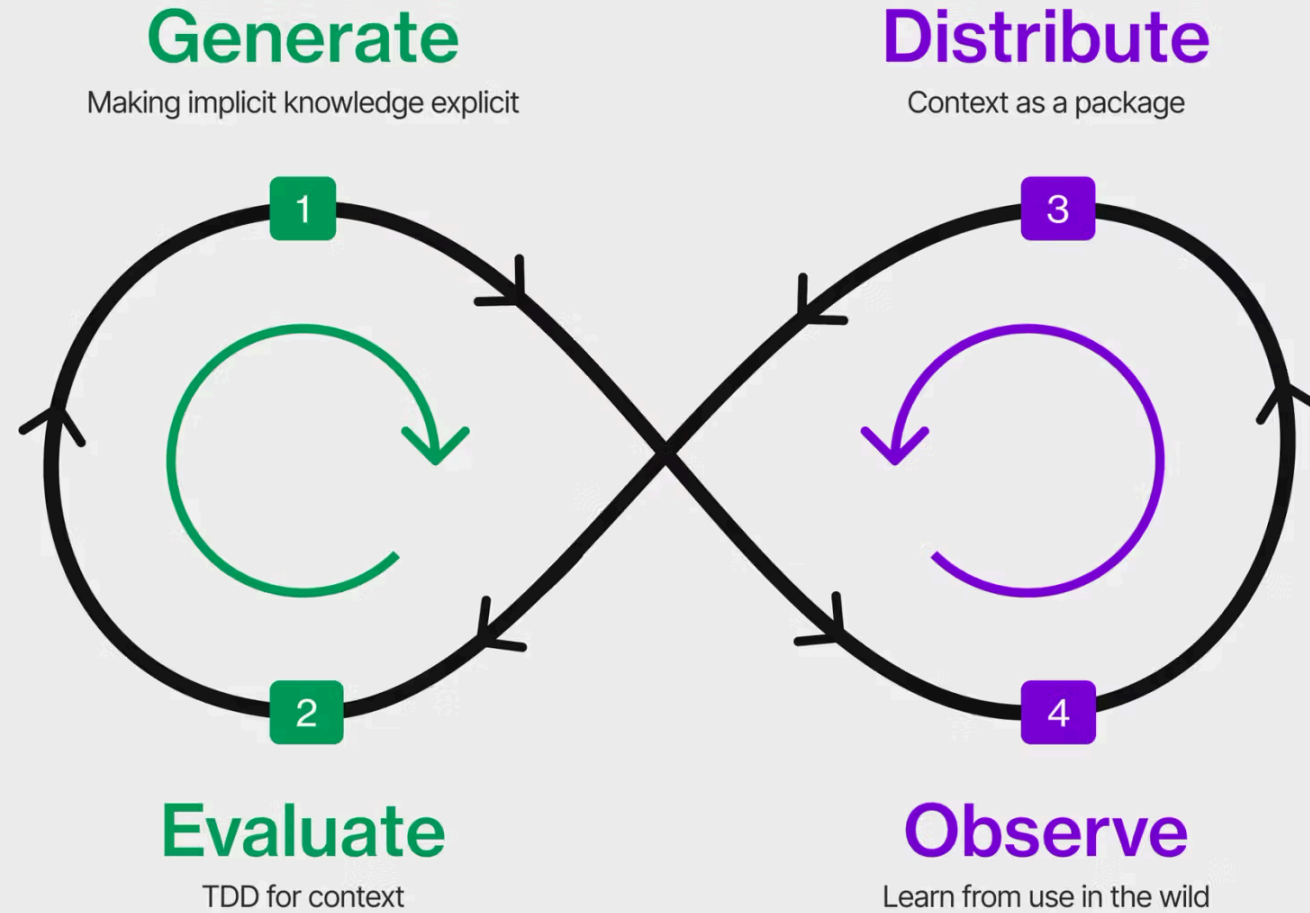
**SDLC**

Software Development Lifecycle

**CDLC**

Context Development Lifecycle

# Context Development Lifecycle



# Today's Agenda

- 01 ✨ Generate — create & curate context
- 02 🔍 Evaluate — test & measure context quality
- 03 📦 Distribute — package & share context
- 04 👁️ Observe — monitor & improve in production



 **Generate**

# Prompts - Humans as context engine

Claude Code v2.1.81

Welcome back Patrick!



Opus 4.6 (1M context) · Claude Team · Tesla AI  
Limited

~/dev/agentics-day

**Tips for getting started**

Ask Claude to create a new ...

**Recent activity**

No recent activity

> I'm going to Agentics Day at Kubecon in Amsterdam. I'm interested in Coding with AI.

Here's the schedule – <https://colocatedeventseu2026.sched.com/overview/area/Agentics+Day%3A+MCP+%2B+Agents>

Tell me the top 3 talks to attend. Avoid generic Kubernetes, infra centric talks.█

<https://colocatedeventseu2026.sched.com/overview/area/Agentics+Day%3A+MCP+%2B+Agents>

# Rules, Instructions - Agent.md

## AGENTS.md

A simple, open format for guiding coding agents, used by over [60k open-source projects](#).

Think of AGENTS.md as a **README for agents**: a dedicated, predictable place to provide the context and instructions to help AI coding agents work on your project.

[Explore Examples](#)

[View on GitHub](#)

```
# AGENTS.md
```

```
## Setup commands
```

- Install deps: `pnpm install`
- Start dev server: `pnpm dev`
- Run tests: `pnpm test`

```
## Code style
```

- TypeScript strict mode
- Single quotes, no semicolons
- Use functional patterns where possible

## Why AGENTS.md?

README.md files are for humans: quick starts, project descriptions, and contribution guidelines.

AGENTS.md complements this by containing the extra, sometimes detailed context coding agents need: build steps, tests, and conventions that might clutter a README or aren't relevant to human contributors.

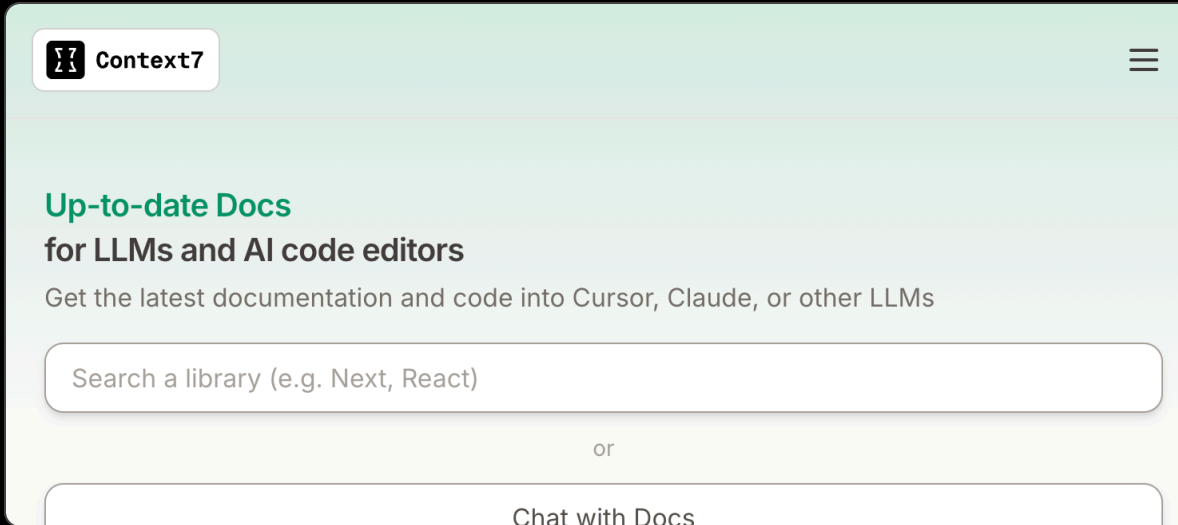
We intentionally kept it separate to:

- 📖 Give agents a clear, predictable place for instructions.
- 👤 Keep READMEs concise and focused on human contributors.
- 🔗 Provide precise, agent-focused guidance that complements existing README and docs.

Rather than introducing another proprietary file, we chose a name and format that could work for anyone. If you're building or using coding agents and find this helpful, feel free to adopt it.

Agent.md

# Context from code and docs



Context7

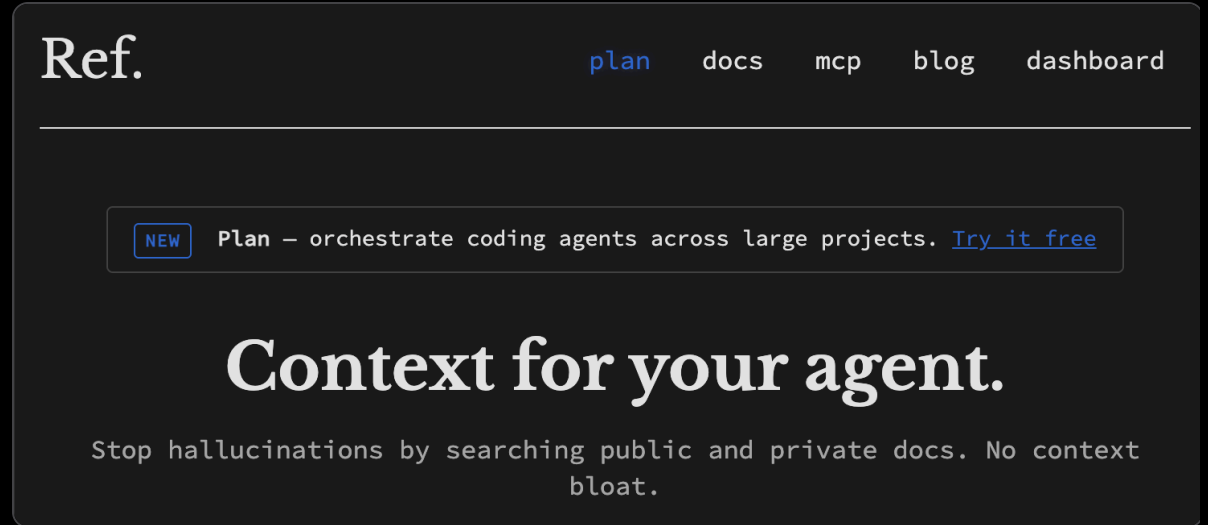
## Up-to-date Docs for LLMs and AI code editors

Get the latest documentation and code into Cursor, Claude, or other LLMs

or

Chat with Docs

<https://context7.com/>



Ref.

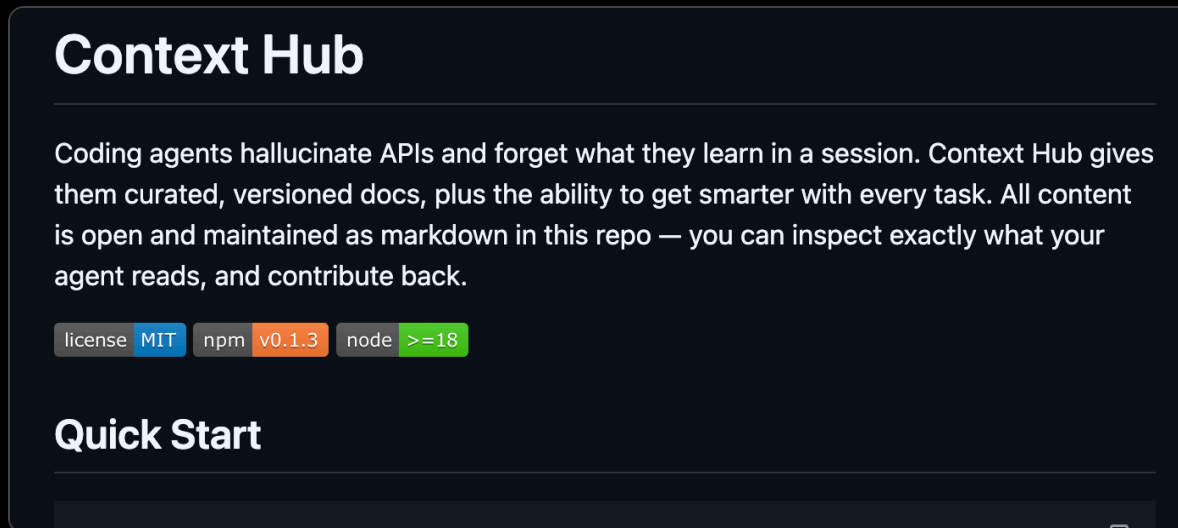
[plan](#) [docs](#) [mcp](#) [blog](#) [dashboard](#)

**NEW** Plan – orchestrate coding agents across large projects. [Try it free](#)

## Context for your agent.

Stop hallucinations by searching public and private docs. No context bloat.

<https://ref.tools/>



## Context Hub

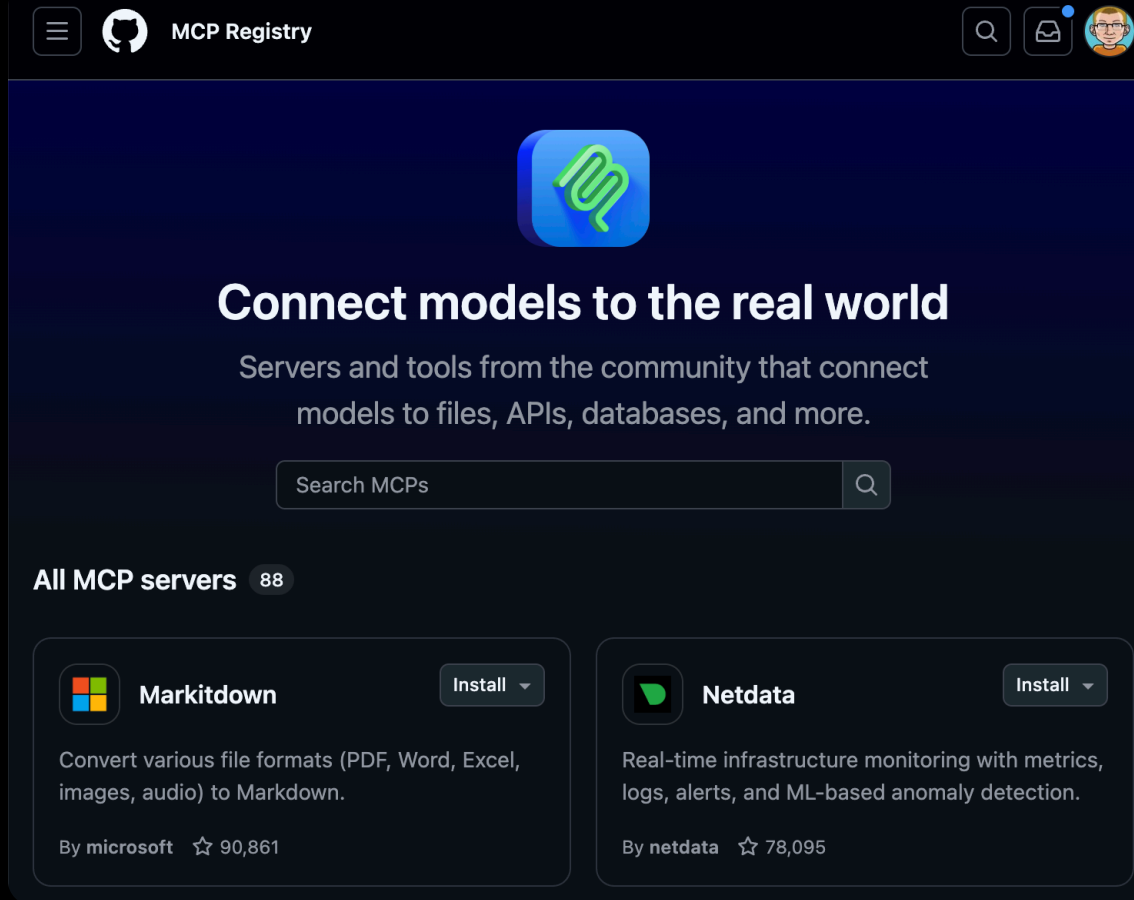
Coding agents hallucinate APIs and forget what they learn in a session. Context Hub gives them curated, versioned docs, plus the ability to get smarter with every task. All content is open and maintained as markdown in this repo — you can inspect exactly what your agent reads, and contribute back.

license MIT npm v0.1.3 node >=18

### Quick Start

<https://github.com/andrewyng/context-hub>

# Context Connectors



MCP Registry

Connect models to the real world

Servers and tools from the community that connect models to files, APIs, databases, and more.

Search MCPs

All MCP servers 88

**Markdown** Install

Convert various file formats (PDF, Word, Excel, images, audio) to Markdown.

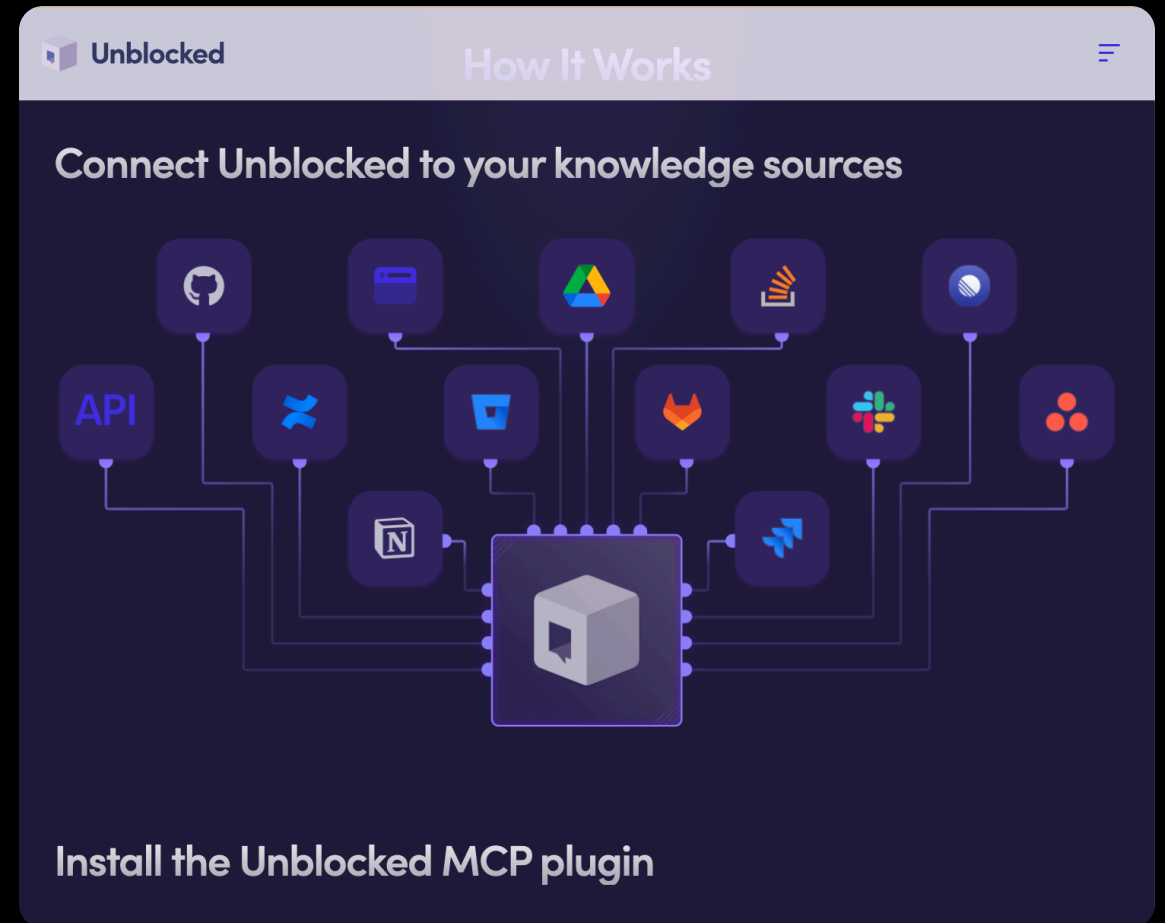
By microsoft ☆ 90,861

**Netdata** Install

Real-time infrastructure monitoring with metrics, logs, alerts, and ML-based anomaly detection.

By netdata ☆ 78,095

Github MCP - <https://github.com/mcp>



Unblocked

How It Works

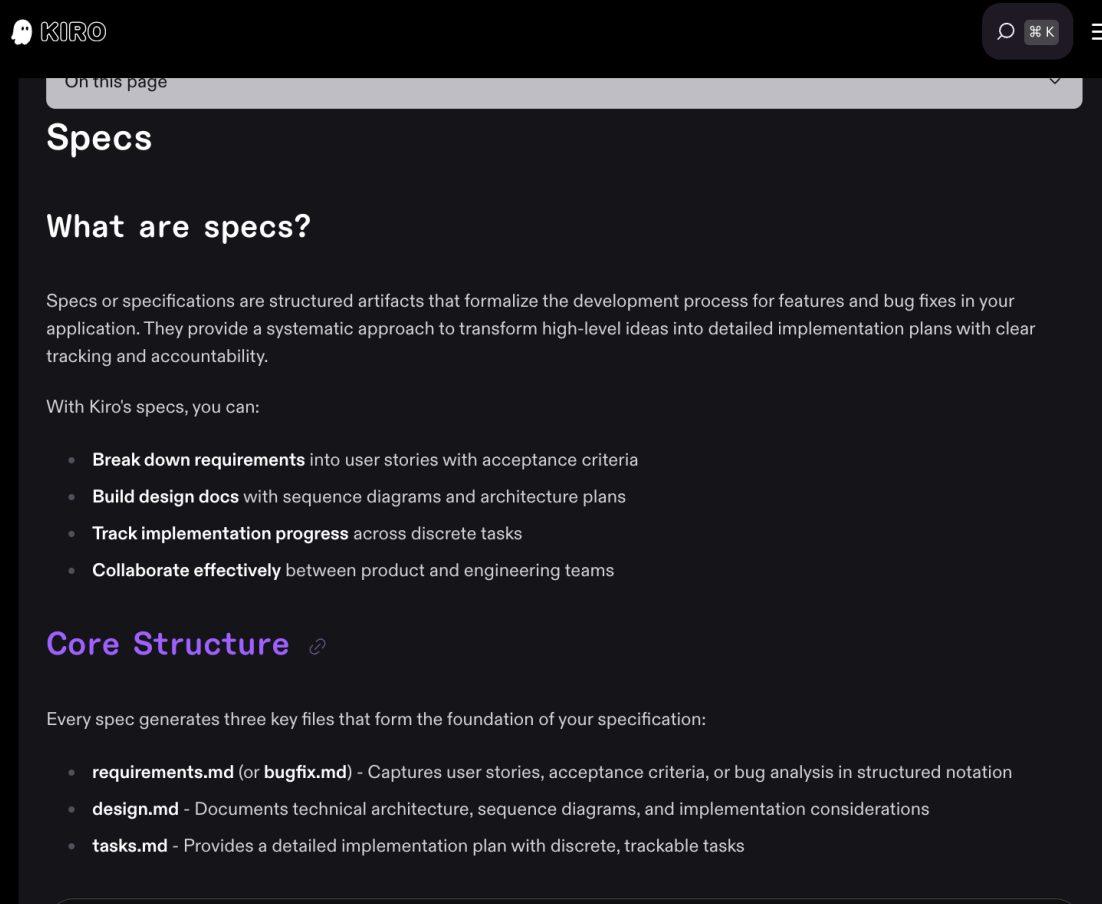
Connect Unblocked to your knowledge sources

API, GitHub, Database, Analytics, Search, etc.

Install the Unblocked MCP plugin

<https://getunblocked.com/unblocked-mcp/>

# Spec Driven Development



The screenshot shows the 'Specs' page on the KIRO website. The page has a dark theme and includes a search bar and a menu icon in the top right. The main heading is 'Specs', followed by a sub-heading 'What are specs?'. The text explains that specs are structured artifacts that formalize the development process. Below this, there is a list of benefits: breaking down requirements, building design docs, tracking implementation progress, and collaborating effectively. A section titled 'Core Structure' lists three key files generated by every spec: requirements.md, design.md, and tasks.md.

KIRO

On this page

## Specs

### What are specs?

Specs or specifications are structured artifacts that formalize the development process for features and bug fixes in your application. They provide a systematic approach to transform high-level ideas into detailed implementation plans with clear tracking and accountability.

With Kiro's specs, you can:

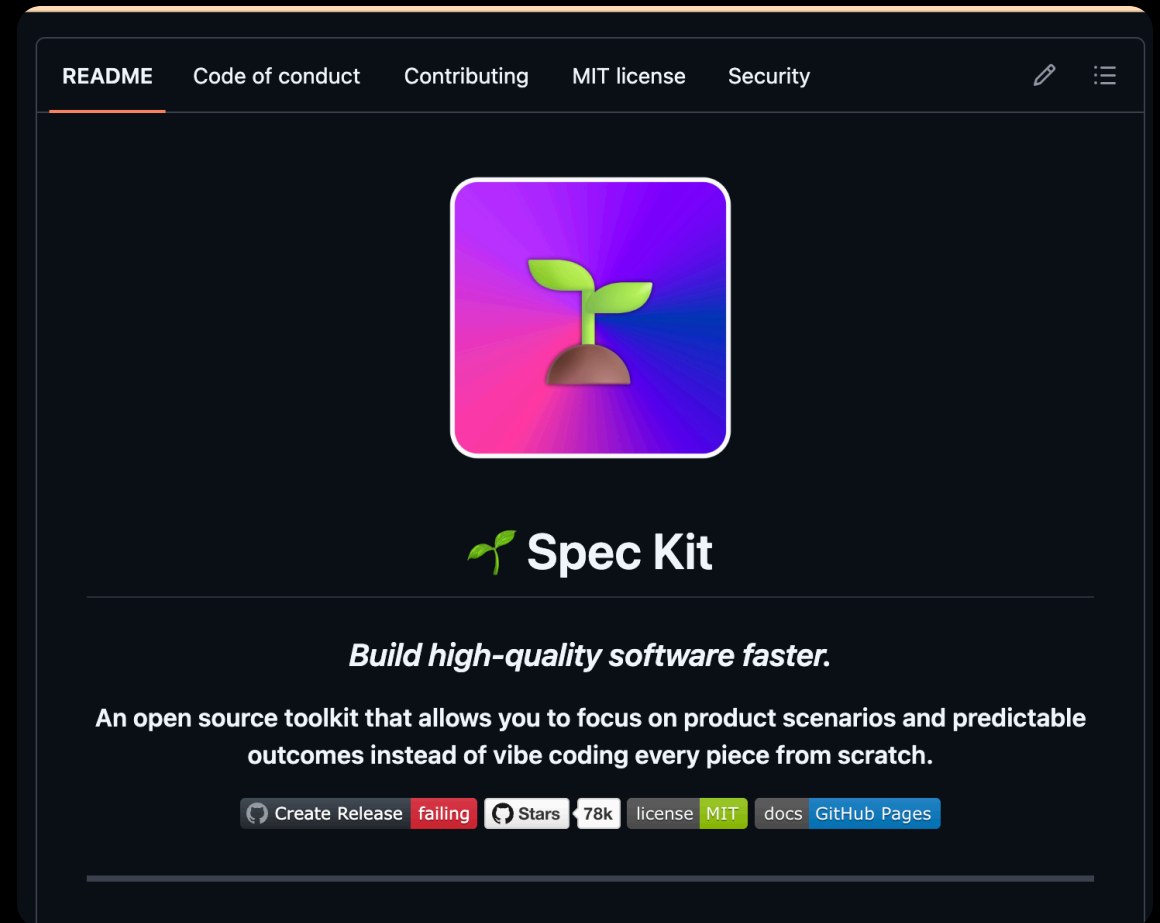
- **Break down requirements** into user stories with acceptance criteria
- **Build design docs** with sequence diagrams and architecture plans
- **Track implementation progress** across discrete tasks
- **Collaborate effectively** between product and engineering teams

### Core Structure

Every spec generates three key files that form the foundation of your specification:


- **requirements.md** (or **bugfix.md**) - Captures user stories, acceptance criteria, or bug analysis in structured notation
- **design.md** - Documents technical architecture, sequence diagrams, and implementation considerations
- **tasks.md** - Provides a detailed implementation plan with discrete, trackable tasks

Kiro - <http://kiro.dev/docs/specs/>



The screenshot shows the GitHub repository page for 'Spec Kit'. The page has a dark theme and includes a navigation bar with links for README, Code of conduct, Contributing, MIT license, and Security. The main heading is 'Spec Kit', followed by a sub-heading 'Build high-quality software faster.' The text explains that it is an open source toolkit that allows you to focus on product scenarios and predictable outcomes instead of vibe coding every piece from scratch. Below this, there is a list of repository statistics: Create Release, failing, Stars, 78k, license MIT, docs GitHub Pages.

README Code of conduct Contributing MIT license Security



## Spec Kit

*Build high-quality software faster.*

**An open source toolkit that allows you to focus on product scenarios and predictable outcomes instead of vibe coding every piece from scratch.**

Create Release failing Stars 78k license MIT docs GitHub Pages

Gitub Speckit - <https://github.com/github/spec-kit>

# Today's Agenda

- 01  Generate — create & curate context
- 02  Evaluate — test & measure context quality
- 03  Distribute — package & share context
- 04  Observe — monitor & improve in production



**Evaluate**

# Skill Markdown

SKILL.md Quality Evals Security

## Terraform Style Guide

Generate and maintain Terraform code following HashiCorp's official style conventions and best practices.

Reference: [HashiCorp Terraform Style Guide](#)

## Code Generation Strategy

When generating Terraform code:

1. Start with provider configuration and version constraints
2. Create data sources before dependent resources
3. Build resources in dependency order
4. Add outputs for key resource attributes
5. Use variables for all configurable values

## File Organization

File

Purpose

## Example Structure

```
# terraform.tf
terraform {
  required_version = ">= 1.7"

  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# variables.tf
variable "environment" {
  description = "Target deployment environment"
  type        = string

  validation {
    condition = contains(["dev", "staging", "prod"], var.environment)
    error_message = "Environment must be dev, staging, or prod."
  }
}

# locals.tf
locals {
  common_tags = {
    Environment = var.environment
    ManagedBy   = "Terraform"
  }
}

# main.tf
resource "aws_vpc" "main" {
  cidr_block      = var.vpc_cidr
  enable_dns_hostnames = true
}
```

<https://docs.tessl.io/evaluate/evaluating-skills>

<https://docs.tessl.io/evaluate/evaluating-skills>

# Skill review ~ Linting

Validation **100%**

Warnings & errors only

Checks the skill against the spec for correct structure and formatting. All validation checks must pass before discovery and implementation can be scored.

Validation — 11 / 11 Passed

Validation for skill structure



Criteria	Description	Result
skill_md_line_count	SKILL.md line count is 354 (<= 500)	Pass
frontmatter_valid	YAML frontmatter is valid	Pass
name_field	'name' field is valid: 'terraform-style-guide'	Pass
description_field	'description' field is valid (165 chars)	Pass
compatibility_field	'compatibility' field not present (optional)	Pass
allowed_tools_field	'allowed-tools' field not present (optional)	Pass
metadata_version	'metadata' field not present (optional)	Pass
metadata_field	'metadata' field not present (optional)	Pass
license_field	'license' field not present (optional)	Pass
frontmatter_unknown_keys	No unknown frontmatter keys found	Pass

<https://docs.tessl.io/evaluate/evaluating-skills>

# Skill eval ~ Grammarly

## Discovery 75%

Based on the skill's description, can an agent find and select it at the right time? Clear, specific descriptions lead to better discovery.

This is a solid description with clear 'what' and 'when' clauses and good distinctiveness for Terraform-specific work. The main weaknesses are moderate specificity (lacks concrete action examples) and limited trigger term coverage (missing common user phrases like 'IaC' or '.tf files').


## Suggestions

Add specific concrete actions like 'create modules, define resources, configure providers, set up backends' to improve specificity

Expand trigger terms to include common variations: 'infrastructure as code', 'IaC', '.tf files', 'terraform modules', 'cloud infrastructure'

Dimension	Reasoning	Score
Specificity	Names the domain (Terraform HCL) and mentions style conventions and best practices, but doesn't list specific concrete actions like 'create modules', 'define resources', 'configure providers', or 'set up state backends'.	2 / 3
Completeness	Clearly answers both what ('Generate Terraform HCL code following HashiCorp's official style conventions and best practices') and when ('Use when writing, reviewing, or generating Terraform configurations') with explicit trigger guidance.	3 / 3

# Task eval ~ Unit Tests

 ☰

## Evaluation results

**92%** ↑ 47% Details

### Project Proposal Document

docx-js document creation for new documents

Criteria	Without context	With context
Uses docx-js library ⓘ	❌ 0%	✅ 100%
Proper heading hierarchy ⓘ	🟡 50%	🟡 60%
Content completeness ⓘ	✅ 100%	✅ 100%
Exports valid .docx file ⓘ	🟡 50%	✅ 100%

<https://tessl.io/registry/skills/github/anthropics/skills/docx/evals>

## Project Proposal Document

docx-js document creation for new documents

✕

# Project Proposal Document

## Context

A project manager at "Cascade Innovations" needs a professional project proposal document for a new product launch. The document should be well-structured with clear sections and a polished appearance suitable for executive review.

## Task

Create a Word document (.docx) for the "Project Atlas" product launch proposal containing the following sections:

- **Title:** "Project Atlas: Product Launch Proposal" centered at the top
- **Project Overview:** 2-3 paragraphs describing the initiative to launch a B2B solution

<https://tessl.io/registry/skills/github/anthropics/skills/docx/evals>

# With your Repo ~ E2E Tests

## scenario.json

Generated by `tessl scenario download`. Defines the fixture for the eval run.

```
{
  "type": "coding",
  "fixture": {
    "type": "commit",
    "repoUrl": "https://github.com/org/repo.git",
    "ref": "24829180ba1fafb86b...",
    "exclude": ["*.mdc", "*.md", "tile.json", "tessl.json", ".tessl/"]
  }
}
```

- `fixture.ref` — the parent commit hash (the starting state for the agent)
- `fixture.exclude` — context patterns stripped for baseline; also used as the default `--context-pattern` at run time
- `fixture.repoUrl` — full clone URL

## task.md

Free-form markdown. This is the **only file the agent sees** — it has no access to `criteria.json`. Typically structured with Problem, Expected Behavior, and Acceptance Criteria sections. You can edit this freely before running.

<https://docs.tessl.io/evaluate/evaluating-your-codebase>

# Skill optimize ~ Code Actions

zack-anthropic skill-creator: drop ANTHROPIC\_API\_KEY requirem... b0cbd3d · 2 weeks ago

485 lines (327 loc) · 32.4 KB

Preview Code Blame

name	description
skill-creator	Create new skills, modify and improve existing skills, and measure skill performance. Use when users want to create a skill from scratch, edit, or optimize an existing skill, run evals to test a skill, benchmark skill performance with variance analysis, or optimize a skill's description for better triggering accuracy.

## Skill Creator

A skill for creating new skills and iteratively improving them.

At a high level, the process of creating a skill goes like this:

- Decide what you want the skill to do and roughly how it should do it

<https://github.com/anthropics/skills/blob/main/skills/skill-creator/SKILL.md>

```
[$ tessel skill review --optimize ./arduin skill  
: Reviewing and improving skill (this can take up to 1
```

A summary of changes will be provided, alongside the improvement score

```
### Best Practices  
+  
+- **Use meaningful variable names**: `ledPin` instead of `p1`  
+- **Add comments**: Explain non-obvious logic for future reference  
+- **Avoid blocking code**: Don't use long `delay()` calls in loops; use `millis()` instead  
+- **Test incrementally**: Upload and verify each feature before adding the next  
+- **Check library documentation**: Many sensors have example code in their Arduino libr
```

### Summary of Changes







Removed the introductory sentence "This skill helps you create functional Arduino sketch the workflow, improving conciseness. Trimmed inline code comments that explain obvious initialize serial communication" after Serial.begin(). Retained the Board Reference tab inline to preserve actionability and reference value, as they are concise and frequently ment. These changes address the verbosity feedback while maintaining the skill's high ac ity scores.

Score: 33% → 96% (+63%) after 3 iterations

You will be prompted to accept the changes, unless you have used the add automatically do so.

<https://docs.tessel.io/evaluate/optimize-a-skill-using-best-practices>

# CI/CD and Evals

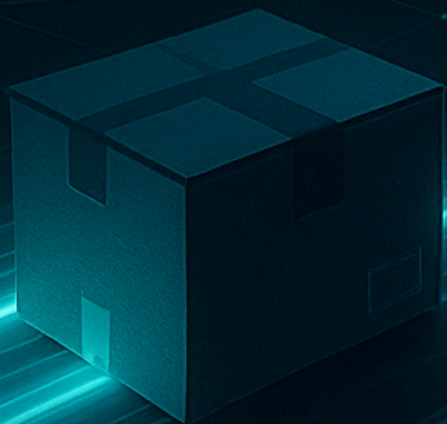
-  Non-deterministic  $\neq$  untestable — think error budgets
-  Run 5+ trials; force binary decisions, not fuzzy scores
-  Fast subset locally  $\rightarrow$  full suite on CI  $\rightarrow$  scheduled drift checks
-  Mine production failures — they're your best eval data
-  Every context change reruns the suite — no exceptions
-  Vendor metrics lie — define your own

# Today's Agenda

- 01  Generate — create & curate context
- 02  Evaluate — test & measure context quality
- 03  Distribute — package & share context
- 04  Observe — monitor & improve in production



**Distribute**



# Share on github

The screenshot shows a GitHub repository page for 'anthropics / skills'. The commit history shows a commit by 'ant-andi' titled 'Move example skills into dedicated folder and create min...' with commit hash 'ef74077' and a timestamp of '3 months ago'. The file 'SKILL.md' is selected, showing 405 lines (302 loc) and 19.3 KB. The 'Preview' tab is active, displaying a table with the following content:

name	description	license
algorithmic-art	Creating algorithmic art using p5.js with seeded randomness and interactive parameter exploration. Use this when users request creating art using code, generative art, algorithmic art, flow fields, or particle systems. Create original algorithmic art rather than copying existing artists' work to avoid copyright violations.	Complete terms in LICENSE.txt

Algorithmic philosophies are computational aesthetic movements that are then expressed through code. Output .md files (philosophy), .html files (interactive viewer), and .js files (generative algorithms).

This happens in two steps:

1. Algorithmic Philosophy Creation (.md file)
2. Express by creating p5.js generative art (.html + .js files)

Anthropic Example Skill - <https://github.com/anthropics/skills/blob/main/skills/algorithmic-art/SKILL.md>

# Package managers & Versioning

## Why APM

AI coding agents need context to be useful — standards, prompts, skills, plugins — but today every developer sets this up manually. Nothing is portable nor reproducible. There's no manifest for it.

**APM fixes this.** Declare your project's agentic dependencies once in `apm.yml`, and every developer who clones your repo gets a fully configured agent setup in seconds — with transitive dependency resolution, just like npm or pip.

```
# apm.yml - ships with your project
name: your-project
version: 1.0.0
dependencies:
  apm:
    # Skills from any repository
    - anthropics/skills/skills/frontend-design
    # Plugins
    - github/awesome-copilot/plugins/context-engineering
    # Specific agent primitives from any repository
    - github/awesome-copilot/agents/api-architect.agent.md
    # A full APM package with instructions, skills, prompts, hooks...
    - microsoft/apm-sample-package
```

```
git clone <org/repo> && cd <repo>
```

APM - <https://github.com/microsoft/apm>

Tessl Docs

Website Press Kit

That directory is linked to `tessl.json`. This tile provides documentation for the FastAPI framework, which you, or your agent, can now query using the `query_library_docs` tool when you need information about FastAPI usage, features, and best practices.

Inside the tile directory, you'll find a manifest file called `tile.json` that contains metadata about the tile, including the package version:

```
{
  "name": "tessl/pypi-fastapi",
  "version": "0.116.0",
  "docs": "docs/index.md",
  "describes": "pkg:pypi/fastapi@0.116.1",
  "summary": "FastAPI framework, high performance, easy to learn, fast",
  "private": true
}
```

- If a tile describes a package, you must have (at a minimum)

- `describes` - a valid PURL to identify the package

<https://docs.tessl.io/use/make-your-agents-smarter-with-documentation>

# Marketplaces & Registries



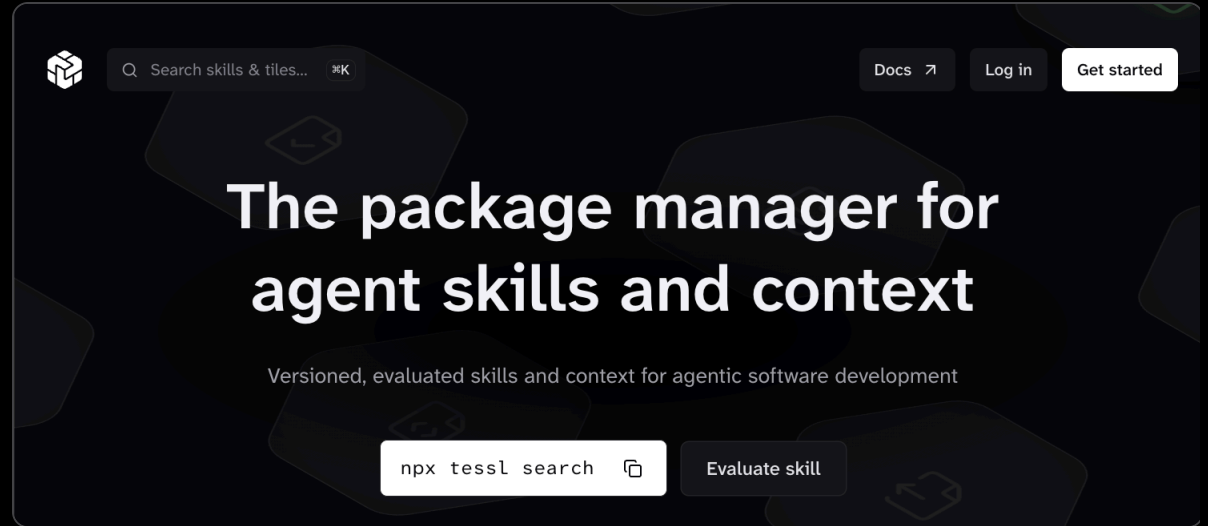
▲ / Skills Curated **NEW** Audits Docs

# SKILLS

THE OPEN AGENT SKILLS ECOSYSTEM

Skills are reusable capabilities for AI agents. Install them with a single command to enhance your agents with access to procedural knowledge.

Skills.sh - <https://skills.sh>



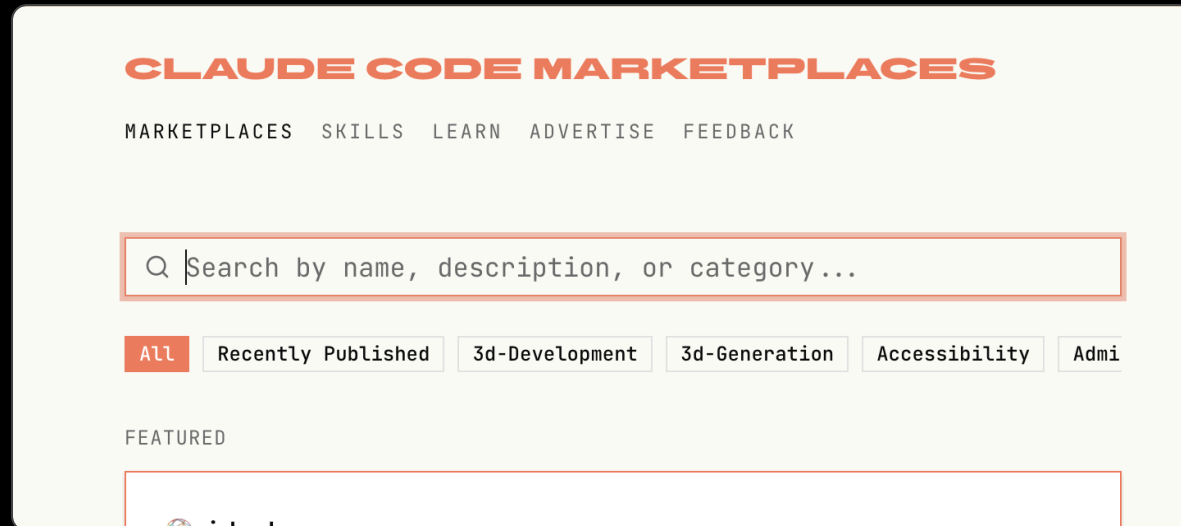
🔍 Search skills & tiles... \*K Docs ↗ Log in Get started

# The package manager for agent skills and context

Versioned, evaluated skills and context for agentic software development

npx tessl search 🔗 Evaluate skill

Tessl Registry - <https://tessl.io/registry>



## CLAUDE CODE MARKETPLACES

MARKETPLACES SKILLS LEARN ADVERTISE FEEDBACK

🔍 Search by name, description, or category...

**ALL** Recently Published 3d-Development 3d-Generation Accessibility Admi

FEATURED

Claude Marketplaces - <https://claudemarketplaces.com/>

# Skills - context package standard ?

## Agent Skills

What are skills?

### What are skills?

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

At its core, a skill is a folder containing a `SKILL.md` file. This file includes metadata ( `name` and `description` , at minimum) and instructions that tell an agent how to perform a specific task. Skills can also bundle scripts, templates, and reference materials.


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

### How skills work

Skills use **progressive disclosure** to manage context efficiently:

<https://agentskills.io/what-are-skills>

# Context security scan



 Agent Scan Skill Inspector

By submitting your skill for analysis, you agree to our [terms of service](#).

**slack** 2 files

Use when you need to control Slack from Clawdbot via the slack tool, including reacting to messages or pinning/unpinning items in Slack channels or DMs.

v1.0.0 by steipete

- >  SKILL.md
- >  \_meta.json

**No issues found**  
9 security checks completed

<input checked="" type="checkbox"/> Prompt Injection	<input checked="" type="checkbox"/> Malicious Code	<input checked="" type="checkbox"/> Suspicious Downloads
<input checked="" type="checkbox"/> Improper Credential Handling	<input checked="" type="checkbox"/> Secret Detection	<input checked="" type="checkbox"/> Third-Party Content Exposure
<input checked="" type="checkbox"/> Unverifiable Dependencies	<input checked="" type="checkbox"/> Direct Money Access	<input checked="" type="checkbox"/> Modifying System Services

# Context provenance

📖 README MIT license

## AISBOM - AI Software Bill of Materials

JSON Spec for Transparency Obligations of the EU AI Act, including LLM / foundation models

Version 0.1 (December 11, 2023)

**Note**

- This JSON file is intended as a means to address the transparency requirements in the upcoming EU AI Act (focus on Article 13 & 52).
- The file is an illustrative example as the basis for discussion and feedback.
- To use the file, copy the template and insert the values of the AI System at hand, using the descriptions given in the template as a guidance).
- The file is not a formal JSON Schema, but we may adopt the schema in the future for improved automated processing.

### Call to action

- Please share your feedback in [GitHub Discussions](#).
- See the call for contributions at the end of this document.

<https://github.com/aai-institute/AI-SBOM>

```
$ git-ai blame src/auth.ts
a731df7 Lois Tam      1) export async function login(email, password) {
a731df7 Lois Tam      2)   const user = await db.findByEmail(email)
a731df7 Lois Tam      3)   if (!user) throw new AuthError('not found')
b9c4e22 Cursor | Opus 4 4)   const valid = await bcrypt.compare(
b9c4e22 Cursor | Opus 4 5)     password, user.passwordHash
b9c4e22 Cursor | Opus 4 6)   )
b9c4e22 Cursor | Opus 4 7)   if (!valid) throw new AuthError('bad creds')
a731df7 Lois Tam      8)   return issueSession(user)
a731df7 Lois Tam      9) }
```

## AI Blame

Git AI links each line of AI-code to the agent, model and prompt that generated it, so that agents and their agents understand the "why" behind every line.

<https://usegitai.com/>

# Today's Agenda


- 01  Generate — create & curate context
- 02  Evaluate — test & measure context quality
- 03  Distribute — package & share context
- 04  Observe — monitor & improve in production



# 4.Observe





# Standardized Agent logs


 Cognition MENU +


## Agent Trace Spec


Version: 0.1.0 • Status: RFC • Date: January 2026 • License: CC BY 4.0


 CURSOR


 Cognition


 CLOUDFLARE

 Vercel

 git-ai

 opencode

 jules

 aillip

**The Central Problem: Throwing Away Context**

Foundation Capital recently wrote a viral piece on [Context Graphs](#) that they define as:

**“a living record of decision traces stitched across entities and time** so precedent becomes searchable. Over time, that context graph becomes the real source of truth for autonomy – because it explains not just what happened, but why [it happened].”

The Central Problem: Throwing Away Context +

<https://cognition.ai/blog/agent-trace>

Announcing Entire with \$60m seed round >

# Every commit tells a story. Now you can read it.

Entire CLI hooks into your git workflow to capture AI agent sessions on every push. Sessions are indexed alongside commits, a searchable record of how code was written.

curl ▾

```
curl -fsSL https://entire.io/install.sh | bash
```



Open source · MIT licensed | ★ 3.6k

entire / contributions

# Context from production logs

{Hud

Blog Docs About us

Book a demo

Log in ↗

## Where code meets reality

Install once. No config. Zero maintenance.

### Understand code behavior with function level data

Hud gathers errors and performance data at the service and function level. It connects, at runtime, the business impact and the root cause in the code. Engineers use this data in the IDE to understand how their code behaves in reality.

The screenshot displays an IDE interface with a code editor at the top and a HUD (Hud User Dashboard) overlay at the bottom. The code editor shows a JavaScript function with a call to `api.post`. The HUD overlay shows a performance summary for the selected function: `10,874/d ↗ 68.94ms 27.50%err ↑`. Below this, a 'Call Graph' section shows 8 callers, with a central node labeled 'You are here' and arrows pointing to various functions like `MetricsClickhouseUtils.form...`, `writeVarUInt(buf, value, offs..`, `definedSketch.data.reduce..`, `PgClient.query(params, con...`, `result.reduce() callback`, and `result.find() callback`.

<https://www.hud.io/>

# Agent sandboxing

An open taxonomy and scoring framework for evaluating AI agent sandboxes. It decomposes sandboxing into **7 defense layers**, maps them against **7 threat categories**, and scores each mechanism on **3 dimensions** (strength, granularity, portability), producing comparable fingerprints for any product. Includes score cards for 23 sandbox tools, a composition framework for stacking complementary products, and a decision checklist for choosing the right sandbox stack.

### The Agent Sandbox Taxonomy

7 · 7 · 3

7 DEFENSE LAYERS	7 THREATS	3 SCORING DIMENSIONS
L7 Observability & Audit	T1 Data Exfiltration	<b>Strength</b> S: 0-4
L6 Action Governance	T2 Supply Chain	0 None
L5 Credential & Secret Mgmt	T3 Destructive Ops	1 Cooperative
L4 Network Boundary	T4 Lateral Movement	2 Software-enforced
L3 Filesystem Boundary	T5 Persistence	3 Kernel-enforced
L2 Resource Limits	T6 Privilege Escalation	4 Structural
L1 Compute Isolation	T7 Denial of Service	<b>Granularity</b> G: 0-3
		0 None
		1 Binary (on/off)
		2 Allow/Blocklist
		3 Per-resource policy
		<b>Portability</b> tags
		any-os · linux · mac · cloud · docker · k8s · kvm

---

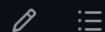
<b>Fingerprint</b> strength per layer	<b>Composition</b> take max per layer
L1/L2/L3/L4/L5/L6/L7	L1/L2/L3/L4/L5/L6/L7
4/4/4/0/2/-/2	4/4/4/0/2/-/2 firecracker vm
	-/-/1/2/3/2/3 network proxy
<b>Score Card</b> adds granularity (S.G)	4/4/4/2/3/2/3 no gaps ✓
4.1/4.2/4.1/0.0/...	

github.com/kajogo777/the-agent-sandbox-taxonomy

<https://github.com/kajogo777/the-agent-sandbox-taxonomy/tree/main>

# Context Filter ~ WAF

📖 README



## context-filter (macOS)

DYLD\_INSERT\_LIBRARIES library to detect and warn about prompt injection in Claude Code instruction files on macOS

### The Problem

When you clone a repository, Claude Code automatically loads `CLAUDE.md` and skill files into context **before any hooks fire**. These files can contain prompt injection attacks that manipulate Claude's behavior.

This library intercepts file reads at the syscall level using `DYLD_INSERT_LIBRARIES`, scans for injection patterns, and prepends a warning to suspicious files—all before the content reaches Node.js/Claude.

#### Two detection layers:

- **Regex patterns** (built-in, always active) — fast pattern matching against known injection signatures based on [Lasso Security's research](#). Configurable via `cf-module/config/patterns.json`.
- **ML scanner sidecar** (optional) — when the scanner daemon is running, the library also sends file content to a prompt injection ML model ([LLM Guard](#) or [NeMo Guardrails](#)) for deeper detection. Falls back to regex-only if the daemon is not running.

Context filter - <https://github.com/jedi4ever/context-filter>

# Today's Agenda

- 01  Generate — create & curate context
- 02  Evaluate — test & measure context quality
- 03  Distribute — package & share context
- 04  Observe — monitor & improve in production

# Unfinished thoughts

- What will chaos engineering look like for context ?
- And context analytics & Context SEO
- How will A/B testing work ?
- Who will be the next Github or Kubernetes for context ?
- Will Agents not do all this autonomously ?
- Will Agents invent their own context language ?



**Context is the fuel.**

Coding agents are the engine

— Patrick Debois

🙏 Thanks for listening !

Connect on LI for the slides send me feedback

📅 June 1 - 2



The conference for developers building real AI-native systems - agents, specs, workflows, and platforms, at production scale.

📍 London and Virtual

GET YOUR PASS

09	06	15	10	44
Weeks	Days	Hours	Minutes	Seconds

Discount Code: PATRICK50