



DCSAI Technologies

TECHNICAL WHITEPAPER · v1.0 · MAY 2026

DCS Agents

A curated catalog of 641 pre-built agents

Install in one click. Forked into your workspace. Same sandbox as your own code.

Abstract

DCS Agents is a curated catalog of 641 pre-built AI agents covering 14 functional categories. Each agent is a single agent.yaml file plus its prompts and test suite. Customers install agents into their workspace with one click; they run inside the DCS Platform runtime with the same sandboxing, capability scoping, and receipt-emission guarantees as agents the customer writes themselves.

The catalog exists for the same reason the App Store exists: most teams do not want to write their own version of every utility. A refund-handler agent is broadly the same problem whether you sell shoes or SaaS — there is room for a well-tested shared implementation. The 641 agents in the catalog are the most-requested generic implementations, contributed by DCS Labs (the in-house research team) and verified third parties.

Agents in the catalog are versioned, reviewed, and signed. Each install creates a fork in the customer's workspace; the customer can modify the fork freely. Upstream updates are opt-in via a clear diff review, never automatic.

Who this is for

Engineering leaders evaluating whether to build or buy AI agents for common workflows. Product managers shopping for a vertical-specific agent (e.g., a healthcare-intake agent). Third-party developers wanting to publish agents to the catalog. Compliance teams reviewing the security posture of installed agents.

Contents

1	Introduction	3
2	Catalog Structure — 14 categories	5
3	Anatomy of an Agent	8
4	Installation + Forking	11
5	Versioning + Updates	14
6	Quality Bar — verified vs. community	16
7	Publisher Program	19
8	Revenue Share	22
9	Security Review	24
10	Performance + Cost	26
11	Comparison vs. OpenAI GPT Store, Anthropic Computer Use	28
12	Implementation Guide	30
13	References	33

1. Introduction

The first generation of agent platforms (LangChain templates, OpenAI GPT Store, Anthropic Computer Use examples) demonstrated that pre-built agents are useful. They also demonstrated three things about what a production catalog needs:

- **Sandboxing.** A catalog agent must run in the same sandbox as a customer-written one. No exceptions.
- **Versioning.** Catalog agents must be versioned; updates must be opt-in; the customer must see the diff before accepting.
- **Receipts.** Catalog agents must emit the same signed receipts as customer agents. The audit trail does not get a special case for "the agent came from the catalog."

2. Catalog Structure



Figure 2.1 — Catalog organized into 14 categories by function.

3. Anatomy of an Agent

A single agent card — what every catalog entry shows

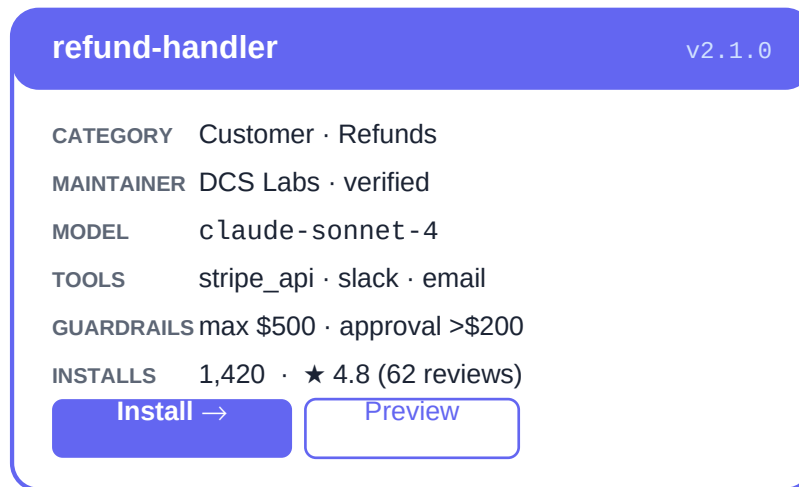


Figure 3.1 — A single catalog card showing version, maintainer, tools, guardrails, popularity.

3.1 The seven mandatory fields

Every agent in the catalog declares:

- **name + version** — semver-like; latest version always installable
- **maintainer** — DCS Labs / verified third-party / community
- **model** — which LLM the agent wraps (catalog agents are model-agnostic where possible)
- **tools** — full MCP capability list; what the agent can touch externally
- **guardrails** — domain-specific limits (dollar caps, approval thresholds, allowed regions)
- **tests** — sample inputs + expected behaviors; must pass before publication
- **license** — Apache-2.0, MIT, or commercial; visible at install time

4. Installation + Forking

Installing a catalog agent creates a fork in the customer's workspace. The customer now owns the fork — they can edit prompts, add tools, tighten guardrails. Changes do not flow back to the catalog unless the customer explicitly contributes them.

```
$ dcs catalog install refund-handler@2.1.0  
forked to workspace · /agents/refund-handler/  
applying default config from .dcs-defaults.yaml ✓  
running install tests ✓  
receipts emitted ✓  
agent active
```

5. Versioning + Updates

When an upstream agent is updated, installed forks get a notification. Customers see the diff (prompts, tools, guardrails) and can accept, reject, or accept-selectively. Updates are never applied automatically. The notification includes a brief changelog explaining what changed and why.

6. Quality Bar

Three tiers of catalog entry exist, distinguished by reviews:

Tier	Reviewer	What's checked	Visible badge
Verified	DCS Labs (paid)	Security · prompts · tests · guardrails · published · 20+ days	✓ Verified (green)
Community	Community review (3 votes)	Security · prompts · tests	★ community
Experimental	Self-published, automated checks only	Static analysis only	■ experimental

7. Publisher Program

Third parties can publish agents to the catalog. The publisher program requires:

- A signed publisher agreement (DCS-issued)
- Identity verification (corporate registration for orgs, government ID for individuals)
- A passing automated security scan on every release
- Public contact details and a response-time commitment for security disclosures

8. Revenue Share

Catalog agents can be free or paid. Paid agents charge an install fee + per-run fee. Revenue split is 70% to the publisher, 30% to DCS. Payouts settle monthly via Stripe Connect.

Tier	Install fee	Per-run fee	Publisher take	DCS take
Free	\$0	\$0	—	—
Pay-per-install	\$1-\$99	\$0	70%	30%
Pay-per-run	\$0	\$0.001-\$1.00	70%	30%
Subscription	\$5-\$500/mo	\$0	70%	30%

9. Security Review

Every Verified-tier agent passes a DCS Labs security review before publication:

- **Prompt injection resistance** — agent is tested against the OWASP LLM Top 10 prompts
- **Capability minimisation** — tools declared must be the minimum needed for the task
- **Guardrail completeness** — every dollar-bearing action has a cap and an approval threshold
- **Test coverage** — declared test suite covers happy path + 5 adversarial cases
- **Output validation** — agent's outputs are checked for PII leakage, allowed-format compliance

10. Performance + Cost

Catalog agents share the same runtime as custom agents, so their performance characteristics are identical (see DCS Platform Whitepaper Chapter 12). The additional cost when running a catalog agent is the publisher's per-run fee (if any), billed alongside the underlying compute + token costs.

11. Comparison vs. Alternatives

	OpenAI GPT Store	Anthropic CUE	LangChain Hub	DCS Agents
Signed agents	X	X	X	✓
Sandboxed runtime	~	~	X	✓
Capability scoping enforced	X	X	X	✓
Per-run cost transparent	~	~	depends on host	✓
Catalog size	~3M	~50	~10k	641 (curated)
Audit trail	X	X	X	✓ (R+1+R+2)
Vendor-neutral models	X	X	✓	✓

The OpenAI GPT Store is much larger but virtually un-curated; most listings are duplicates. DCS Agents trades quantity for verifiability + curation.

12. Implementation Guide

12.1 Browse + install

```
$ dcs catalog search "refund"
Found 12 agents:
✓ refund-handler (v2.1.0)    · DCS Labs · 1,420 installs · ★ 4.8
✓ stripe-refund-pro (v1.4)  · Pintastic · 280 installs · ★ 4.6
■ refund-bot (v0.3)         · community · 18 installs · ★ 3.9
...

$ dcs catalog install refund-handler@2.1.0
```

13. References

- [1] Apple App Store. **App Review Guidelines**. (Curation model reference)
- [2] Anthropic. **Building Effective Agents**. December 2024.
- [3] OpenAI. **GPT Store policies**. 2024.
- [4] OWASP. **OWASP Top 10 for LLM Applications**. 2024.
- [5] GitHub. **Marketplace publisher guidelines**. (Verified-publisher model)

Published under CC BY 4.0. Catalog manifest format open at github.com/dcs-platform/agent-catalog.