CustomGPT plugin proposal

ok here is the deal -- Anthropic launched a marketplace for Claude code.. you can see the details here:

https://code.claude.com/docs/en/discover-plugins#official-anthropic-marketplace

https://code.claude.com/docs/en/discover-plugins

https://code.claude.com/docs/en/plugin-marketplaces

.. now here is the deal: As the #1 provider of RAG solutions and the #1 RAG API, it would be fantastic if CustomGPT.ai can have an app on this official marketplace. You can see our API here: https://customgpt.ai/api and our docs are here: https://docs.customgpt.ai/ -- the goal is to launch an app that would add tremendous value for the developer experience by bringing our industry-leading RAG to the Claude code developer experience. Lets work on this .. I want you to cover all the angles - the PR-FAQ, the benefits for developers, the features, the actual use cases that would be unleashed from having a world-class RAG in the Claude code developer and technical business-user experience. I myself am a hard-core user of Claude code using it 10 hours per day, so would love to see this .. lets do this .. I need you to think like a seasoned product manager and think about both product benefits and distribution and other things that a full-stack builder would think about. Your deliverable will be assets based on all considerations and stakeholders in this. think hard, lets nail this.

Below is a full "product + distribution + build" asset pack to get **CustomGPT.ai** into the **official Claude Code marketplace** as a first-class, high-leverage developer experience upgrade.

I'm grounding this in how **Claude Code plugins + marketplaces** actually work (official marketplace name, install flows, plugin component types, MCP server bundling, hooks, etc.) and how **CustomGPT's Hosted MCP Server** works (SSE endpoint per agent, token model, tool surface like **`send_message`**, optional **`upload_file`**, etc.). Claude Code+3

---

1) The big idea

The problem Claude Code power-users hit

Even with Claude Code's excellent repo understanding, the *highest-friction moments* in real development aren't in the codebase—they're in the "institutional knowledge layer":

- architecture decisions, ADRs, RFCs
- internal runbooks & ops procedures
- security/compliance requirements
- product specs & edge cases
- customer-facing docs & support macros

- “tribal knowledge” living in Notion/Confluence/Drive/PDFs

You can paste snippets into the chat, but it’s slow, lossy, and doesn’t scale.

The solution

A CustomGPT.ai plugin in the official Claude Code marketplace that brings **enterprise-grade RAG** into Claude Code *as tools Claude can call automatically*—so the model can fetch *exact* policy/requirements/runbook context with citations, right when it’s writing code, reviewing diffs, or debugging.

The cleanest implementation path is to leverage **CustomGPT’s Hosted MCP Server** (per agent) and bundle it into a Claude Code plugin via an MCP configuration. Claude Code plugins can bundle MCP servers, which then appear as tools. Claude Code+1

---

2) What “official marketplace” means for distribution

Claude Code has an **official Anthropic marketplace** named **`claude-plugins-official`** that is **automatically installed**. Users can install plugins from it directly:

```
text


/plugin install plugin-name@claude-plugins-official
``` :contentReference[oaicite:2]{index=2}

That’s the distribution jackpot: zero “add marketplace” friction, maximum trust, and immediate discovery through `/plugin → Discover`.

---

3) Product: What we should ship

Product name (working)
CustomGPT RAG for Claude Code (plugin name: `customgpt`)

One-liner
“Instant, cited answers from your company’s knowledge—inside Claude Code—powered by CustomGPT’s Hosted MCP RAG.” :contentReference[oaicite:3]{index=3}

Target users
1) **Hardcore Claude Code developers** (like you): 10 hrs/day, want “no context switching.”
2) **Engineering leads / staff+**: enforce standards, reduce incident time.
3) **Technical business users** working alongside engineers: PM, solutions, support eng.

---

4) MVP feature set (high impact, low complexity)

Claude Code plugins can include **Commands, Agents, Skills, Hooks, MCP servers**.
:contentReference[oaicite:4]{index=4}
The MVP should use:

A) MCP server integration (the core)
- Bundle a small MCP “bridge” that connects Claude Code to **CustomGPT Hosted MCP Server**.
- CustomGPT Hosted MCP exposes an SSE endpoint per agent:

`https://mcp.customgpt.ai/projects/<PROJECT_ID>/sse?token=<TOKEN>` :contentReference[oaicite:5]{index=5}

- The Hosted MCP always includes `send_message` for RAG; advanced plans may include `upload_file`, `list_sources`, etc. :contentReference[oaicite:6]{index=6}

Why this matters: it makes the plugin feel native—Claude can call the tool whenever it needs
```
```

```
grounding context.

### B) Slash commands for explicit workflows
Claude Code plugin commands live in `commands/` and become namespaced like `/plugin-
name:command`. :contentReference[oaicite:7]{index=7}

Recommended commands:

1. `/customgpt:setup`
   - Walks user through: create/open agent → deploy MCP server → generate token → set env vars.
   - Mirrors CustomGPT's documented setup steps (Deploy → MCP Server (Beta) → Generate MCP
Token). :contentReference[oaicite:8]{index=8}

2. `/customgpt:ask <question>`
   - Calls `send_message` via MCP and prints:
     - concise answer
     - citations / sources summary (where available)

3. `/customgpt:use-agent`
   - Switch active agent for this repo (e.g., "engineering", "security", "product-specs")

4. `/customgpt:troubleshoot`
   - Checks token presence, agent id, MCP bridge running, etc.

### C) A "Skill" so Claude can autonomously use RAG
Skills are **model-invoked**: Claude decides when to use them based on context.
:contentReference[oaicite:9]{index=9}

Skill behavior:
- When user asks about internal standards/policies/runbooks/specs, Skill triggers `send_message`
to CustomGPT and returns cited context.

### D) Optional (but powerful) "Hook" for guardrails
Hooks can trigger on events like `PostToolUse`, `SessionStart`, etc.
:contentReference[oaicite:10]{index=10}

MVP hook ideas:
- After Claude edits files (`PostToolUse` matcher `Write|Edit`), run a lightweight "policy
check" prompt that says:
  - "If you changed auth/data-handling/PII, fetch relevant policy from CustomGPT and confirm
compliance."

This turns RAG into a *continuous safety net*.

---

## 5) Use cases that become possible (and feel magical)

### 5.1 "Coding with institutional memory"
**Prompt:** "Implement rate limiting consistent with our platform policy."
Claude (via Skill) pulls the exact internal policy excerpt and applies it to the code, with
citations.

### 5.2 Onboarding: zero-tribal-knowledge ramp
**Prompt:** "How do we deploy this service and what are the rollback steps?"
CustomGPT returns the runbook section; Claude executes steps / proposes commands.

### 5.3 PR review that enforces *your* standards
**Prompt:** "Review this diff against our secure coding guidelines and API style guide."
Hook/command fetches standards, then reviews with grounded criteria.

### 5.4 Incident response + root cause with real context
**Prompt:** "This error is happening in prod—what does the on-call runbook say and what did we
do last time?"
Pull runbook + postmortems + known issues from CustomGPT.

### 5.5 "Spec-to-code" without losing requirements
**Prompt:** "Implement this feature exactly per the PRD and edge cases in the spec."
Claude uses CustomGPT agent connected to PRDs/RFCs, reducing missed requirements.

### 5.6 Support engineering inside the dev workflow
When debugging a customer issue: pull product docs + internal escalation notes while editing
code.

### 5.7 Multi-agent knowledge routing
```

```
Use separate CustomGPT agents:
- `security-policies`
- `platform-architecture`
- `customer-docs`
- `team-runbooks`

Then pick agent per repo or per command.

---

## 6) PR-FAQ (Press Release + FAQ)

### Press Release (draft)

**CustomGPT.ai brings enterprise RAG to Claude Code via the official Anthropic marketplace**

Today, CustomGPT.ai announced **CustomGPT RAG for Claude Code**, a new plugin available through
the **official Anthropic Claude Code marketplace**, enabling developers to access their
organization's trusted knowledge directly within Claude Code.

With the plugin installed, Claude Code can retrieve grounded context from CustomGPT agents—such
as internal architecture docs, runbooks, policies, product specs, and customer documentation—
using **Model Context Protocol (MCP)** tools. The integration is powered by CustomGPT's **Hosted
MCP Server**, which provides an authenticated, streaming MCP endpoint per agent and exposes
tools like `send_message` for retrieval-augmented responses. :contentReference[oaicite:11]
{index=11}

Developers can browse and install plugins from the official marketplace (`claude-plugins-
official`), which is automatically available in Claude Code. :contentReference[oaicite:12]
{index=12}

**Availability:** Starting today, users can install the plugin via Claude Code's plugin manager
and connect it to any CustomGPT agent. CustomGPT's Hosted MCP Server is available on existing
plans, with free-trial users able to test it as well. :contentReference[oaicite:13]{index=13}

---

### FAQ (draft)

**What does the plugin do?**
It lets Claude Code access your CustomGPT agents (your knowledge bases) as MCP tools, so Claude
can retrieve accurate, cited context while coding, reviewing, and debugging.
:contentReference[oaicite:14]{index=14}

**How do users install it?**
From Claude Code, install via the official marketplace:
```text
/plugin install customgpt@claude-plugins-official
```

Then run **`/customgpt:setup`**. Claude Code

How do users connect it to their knowledge?

They enable CustomGPT's Hosted MCP Server for an agent and generate an MCP token in the CustomGPT UI (Deploy → MCP Server (Beta) → Generate MCP Token). CustomGPT

What authentication model is used?

CustomGPT uses a bearer MCP token; it's a secret and should be kept out of repos and rotated if exposed. CustomGPT

What MCP endpoint is used?

CustomGPT exposes an SSE stream endpoint per agent at:

`https://mcp.customgpt.ai/projects/<PROJECT_ID>/sse?token=<TOKEN>` CustomGPT

What tools does the MCP server provide?

You'll always see **`send_message`** (RAG search). Some plans include advanced tools like **`upload_file`**, **`list_sources`**, etc. CustomGPT
```

Can this also help developers integrate CustomGPT into their own apps?
Yes—CustomGPT also hosts a docs MCP server at `https://docs.customgpt.ai/mcp` to help tools like Claude Desktop/Cursor/Windsurf answer "how do I integrate CustomGPT" questions and generate integration code. CustomGPT

What data is shared?
At minimum: the query and whatever context is necessary to answer. If users choose to upload/sync, then additional content may be transmitted. (In the marketplace listing, we should be explicit about what is sent, when, and how to disable uploads.)

Is this enterprise-ready?
The marketplace review will care about security posture, secrets handling, and transparent data flow. We should ship with: no token persistence in repo, clear docs, and opt-in for any write/upload operations.

---

7) Product spec: "CustomGPT for Claude Code" (MVP → v1 → v2)

MVP (2–3 weeks build scope)

Goal: Make CustomGPT agents usable in Claude Code as MCP tools + provide "setup/ask" commands.

Components

- `MCP bridge` (Node) launched from plugin `.mcp.json` (Claude Code plugin supports MCP servers in `.mcp.json` / plugin config). Claude Code
- Commands:
  - `/customgpt:setup`
  - `/customgpt:ask`
  - `/customgpt:use-agent`
  - `/customgpt:troubleshoot`
- Skill:
  - "When internal knowledge needed, call CustomGPT `send_message`."

UX principles

- 60-second time-to-first-answer
- "Explain what you're doing" when calling tools
- Always encourage citations / source transparency

v1 (power user delight)

- Multi-agent routing ("pick best agent based on question")
- Optional hook-based "policy guardrails"
- Convenience command: `/customgpt:search-sources`
- Better citations rendering (source name, page, confidence)

v2 (platform play)

- One-command “index this repo into a CustomGPT agent”
  - gated behind explicit opt-in
  - uses MCP **`upload_file`** if available (or API endpoints) CustomGPT+1
- Team/Org mode:
  - “company agent registry”
  - per-repo default agent config
- Admin analytics (usage, top queries, missing docs)

---

8) Marketplace listing assets (copy you can paste)

Marketplace title

CustomGPT RAG

Short description (≤140 chars)

Bring your company’s knowledge into Claude Code with CustomGPT Hosted MCP RAG—grounded answers with sources while you code. CustomGPT+1

Long description

CustomGPT RAG connects Claude Code to your CustomGPT agents (knowledge bases) through MCP. Once installed, Claude can retrieve accurate context from internal docs (runbooks, specs, policies, wikis, PDFs) while editing code, reviewing diffs, and debugging.

- Powered by CustomGPT Hosted MCP Server (SSE endpoint per agent) CustomGPT
- Adds **`/customgpt:*`** commands for setup and querying
- Includes an optional Skill so Claude can automatically fetch grounding context when needed Claude Code+1

Installation

````
text


/plugin install customgpt@claude-plugins-official
/customgpt:setup
``` :contentReference[oaicite:26]{index=26}

### Tags
`rag`, `knowledge-base`, `docs`, `enterprise`, `mcp`, `compliance`, `oncall`, `runbooks`

---

## 9) Security & trust pack (what Anthropic + enterprise reviewers will look for)

Claude Code’s own plugin system supports powerful integrations (MCP servers, hooks). The
official marketplace emphasizes trust—users should trust plugins before installing.
:contentReference[oaicite:27]{index=27}
````
````

```
So we should proactively ship a reviewer-friendly pack:

### Data flow transparency
- Exactly what leaves the machine:
  - query text
  - optional snippets/metadata
  - never upload repo by default
- When uploads occur:
  - only via explicit `/customgpt:sync` and with confirmation

### Secrets handling
- MCP token treated as bearer secret; never write to repo by default
- Encourage env vars or user-scope config
- Clear rotation instructions (CustomGPT UI token rotation) :contentReference[oaicite:28]
{index=28}

### Permissions & least privilege
- MVP read-only mode by default (`send_message`)
- “Write” tools (`upload_file`) only enabled with explicit opt-in (and obvious UI)

### Safe failure modes
- If MCP not configured: commands degrade gracefully and guide user to `/customgpt:setup`
- If token invalid: show minimal error, no leaking

### Supportability
- `/customgpt:troubleshoot` prints diagnostic checklist (without printing secrets)

---

## 10) Distribution plan (official + “get traction before approval”)

### Track A: Official marketplace inclusion (the goal)
Because `claude-plugins-official` is auto-installed, getting listed there is the highest ROI
channel. :contentReference[oaicite:29]{index=29}

**What we do:**
- Build plugin to marketplace standards (docs, security, stability)
- Provide a clean “partner submission packet”:
  - demo video (2 min)
  - security model
  - why it’s unique vs other knowledge tools
  - metrics we’ll track
  - support commitment

### Track B: Immediate adoption before official listing (recommended)
Claude Code supports adding marketplaces from GitHub repos, URLs, etc.
:contentReference[oaicite:30]{index=30}

So you can ship **today** with:
- a `customgpt/claude-code-marketplace` repo containing marketplace.json + plugin
- users install with:
  - `/plugin marketplace add owner/repo`
  - `/plugin install customgpt@owner-repo`

This lets you:
- iterate quickly on DX
- collect usage + testimonials
- arrive to Anthropic with proof (“X installs, Y weekly active, Z enterprise teams”)

---

## 11) Success metrics (what we should measure)

### Activation & retention
- % installs that complete setup (token + agent configured)
- time-to-first-answer (goal: < 2 minutes)
- weekly active users / weekly tool calls

### Quality
- “citation rate” (% answers with sources)
- user thumbs up/down (if captured)
- “follow-up question rate” (proxy for unclear answers)

### Business
- free → paid conversion (Hosted MCP “zero extra cost” messaging helps)
```

```
:contentReference[oaicite:31]{index=31}
- number of connected agents per org
- expansion: “agent created for engineering docs” → “agent created for product + support”

---

## 12) Concrete demo script (2 minutes, shows the magic)

1) Install plugin
   `/plugin install customgpt@claude-plugins-official` :contentReference[oaicite:32]{index=32}

2) Setup
   `/customgpt:setup`
   (Show CustomGPT UI: Deploy → MCP Server → Generate token) :contentReference[oaicite:33]
{index=33}

3) Ask a real internal question
   “What’s our standard approach to auth token refresh + rotation?”

4) Claude retrieves via CustomGPT MCP (`send_message`) and returns:
   - short answer
   - citations / doc pointers

5) Now apply it to code
   “Implement it in this service and add tests.”

6) Close with:
   “No context switching. No guessing standards. It just knows.”

---

## 13) Recommended “next actions” checklist (do this in order)

### Product + engineering
- [ ] Decide MVP surface: read-only RAG first (recommended)
- [ ] Implement MCP bridge (connect to CustomGPT SSE endpoint) :contentReference[oaicite:34]
{index=34}
- [ ] Add `/customgpt:setup` and `/customgpt:ask`
- [ ] Add a Skill for autonomous retrieval :contentReference[oaicite:35]{index=35}
- [ ] Write “Security & Data Flow” section in README

### Distribution
- [ ] Publish a public marketplace repo for early installs (Track B)
:contentReference[oaicite:36]{index=36}
- [ ] Collect 10–20 strong testimonials from Claude Code daily users
- [ ] Package submission packet for Anthropic official marketplace review

### PR/Marketing
- [ ] Launch post: “RAG inside Claude Code: coding with institutional memory”
- [ ] Short video demo + GIF
- [ ] Co-marketing ask: mention in Claude Code plugin highlights

---

If you want, I can also produce **copy-perfect assets** in the exact formats you’ll submit/use:
- an “Anthropic partner submission” one-pager
- the marketplace listing JSON entry fields (aligned to Claude’s marketplace format)
:contentReference[oaicite:37]{index=37}
- the `customgpt` plugin folder skeleton (`.claude-plugin/plugin.json`, `.mcp.json`,
`commands/*.md`, `skills/*/SKILL.md`) based on Claude Code’s plugin reference
:contentReference[oaicite:38]{index=38}
::contentReference[oaicite:39]{index=39}
```