

# Distributing MCP Servers With OCI To Power Agent Skills

**Bobby House**

Sr. Software Engineer, Docker



# Motivation

An agent skill is a markdown file. It references tools by name. Those tools are provided by MCP servers. That's a dependency — and we need a way to manage it.

## Goals

- **Addressable** — a single immutable reference, no drift
- **Configurable** — configuration must be part of the dependency, set by the skill author or deferred to the user
- **Decoupled** — MCP servers should have their own versioning and release cycle, referenced as dependencies rather than bundled with the skill



## Why not `.mcp.json` ?

The obvious first approach — ship an `.mcp.json` with the skill listing the MCP servers it needs.

### Two problems a dependency shouldn't have

- **It defines runtime** — an execution model, not a dependency declaration
- **Configuration is static** — cannot be deferred to the user



## Why not `server.json`?

The file needs modification — the schema is closer, but has gaps.

- Tags may need resolving to digests (*Addressable*)
- We want to supply or override values (*Configurable*)
- `isRequired`, `default`, `placeholder` ≠ "who provides this value?"

Worth exploring — feels very close.



# Authoring

Let's build a skill. The dependency is a profile — a YAML manifest packaged as an OCI artifact and stored in a registry.



## Example

Invoking skill-builder to create a meditations skill — server dependencies are resolved and packed into an OCI artifact:

```
Use skill-builder

name: meditations

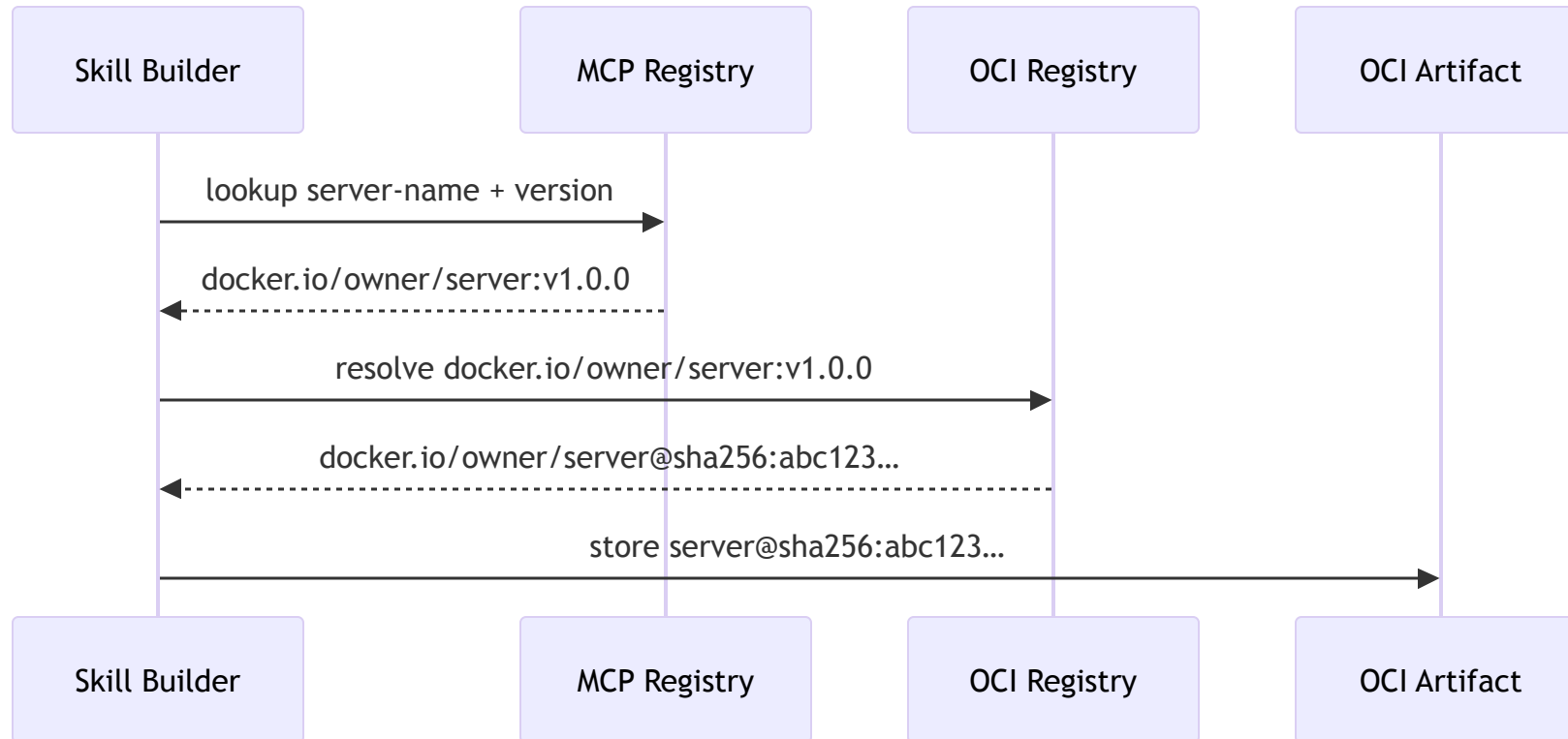
short-description: get a passage from marcus aurelius meditations

details: use io.github.bobbyhouse.project-gutenberg-mcp@1.2.0 and io.github.bobbyhouse.append-log-mcp@1.0.0 to get
passages from meditations and use log to see if we have already seen that passage. If we have already seen that
get another passage, we only care that we haven't seen the same passage within the last 30 days. With the raw pa
text, provide just a small excerpt that includes the original text and a terse interpretation
```



# List of servers

*(Goal: Addressable + Decoupled)*



# Configuration

*(Goal: Configurable)*

Configuration with values either specified by the skill author or explicitly left for the user to supply.

```
1 name: meditations-profile
2 servers:
3   - name: io.github.bobbyhouse/project-gutenberg-mcp
4     identifier: roberthouse224/project-gutenberg-mcp@sha256:6460cba7b27343be72a85cbf5484e024711eb3...
5     config:
6       GUTENBERG_BASE_URL: # undefined - user supplies this
7       GUTENBERG_TOOLS: list_passages,get_passage
8       GUTENBERG_BOOK_ID: "2680"
9   - name: io.github.bobbyhouse/append-log-mcp
10    identifier: roberthouse224/append-log-mcp@sha256:5008d346d8c653caab0309999e694a2a63cf3b2c3bcb7...
11    config:
12      APPEND_LOG_TOOLS: append,query
13      APPEND_LOG_FILE: /data/log.jsonl
```




# Single Dependency



→ mcp-dev-summit-2026 git:(mcp-dev-summit-2026) × claude

Claude Code v2.1.84

Welcome back!



Sonnet 4.6 · API Usage Billing  
~/git/udsrvbtest/talks/mcp-dev-summit-2026

Tips for getting started  
Run /init to create a CLAUDE.md file with instructions for Claude

Recent activity  
No recent activity

/skill-builder:skill-builder

- I'll guide you through building a new Claude Code skill. Let's start with the basics.

Phase 1 – Skill intent

1. What is the skill name? (This becomes the slash command, e.g. my-skill → /my-skill)
2. What does it do? (one sentence)
3. What should it actually accomplish? (detailed intent – used to write the skill body)

[Pasted text #1 +3 lines]

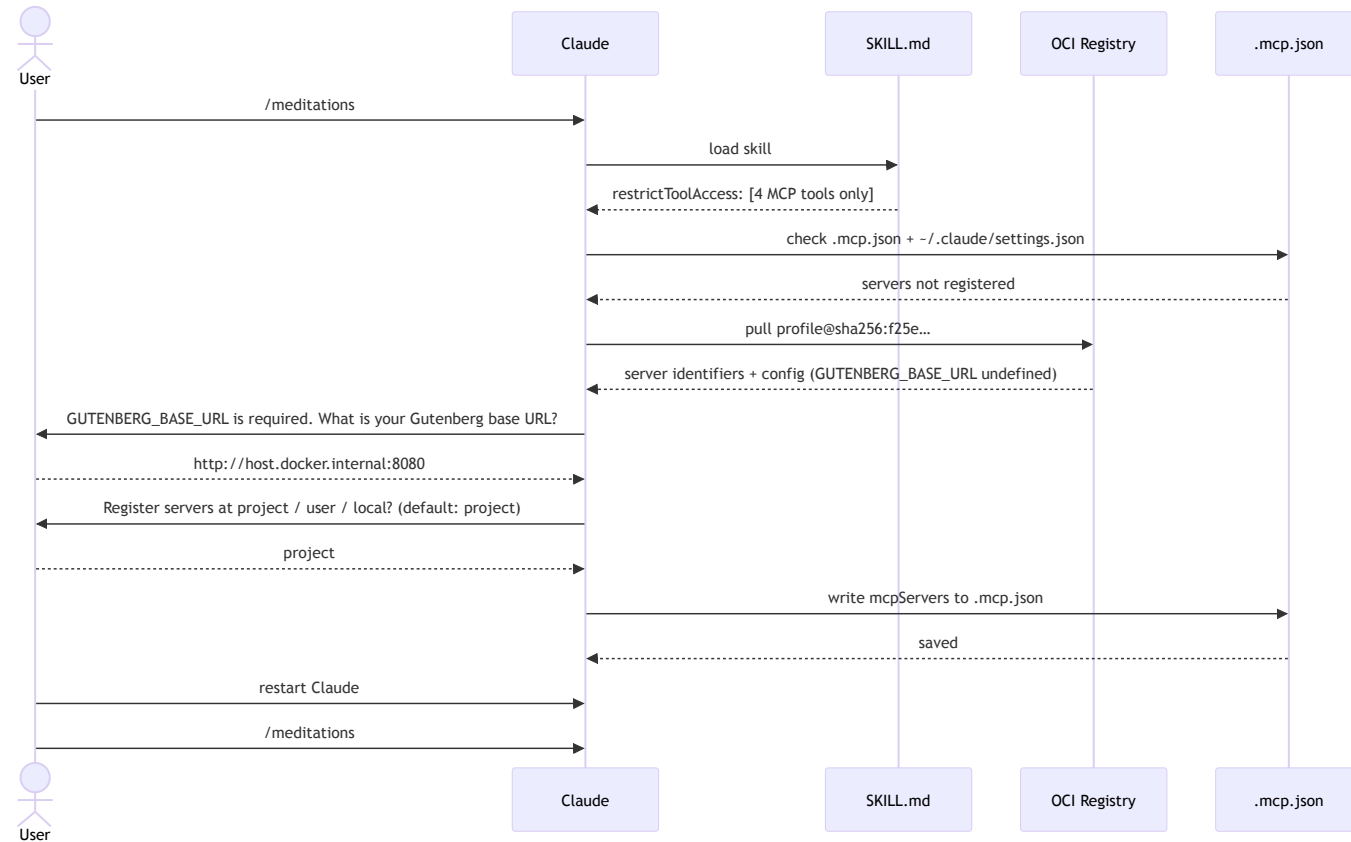
-- INSERT --

▶ 0:00 / 0:55

🔊 🗨️ ⋮

10

# Running the Skill





Recent activity  
No recent activity

Sonnet 4.6 · API Usage Billing  
~/git/udsrvest/talks/mcp-dev-summit-2026

/demos:meditations

Read 1 file (ctrl+o to expand)

- **Bash**(cat ~/.claude/settings.json | python3 -c "import sys,json; d=json.load(sys.stdin); print(json.dumps(list(d.get('mcpServers',{}).keys()), indent=2))" 2>/dev/null...)  
└─ []
  - Neither file has the required servers. Let me set them up – starting by pulling the profile.
  - **Bash**(docker pull docker.io/roberthouse224/meditations-profile@sha256:b6dfc8b5758a9d92a9f5c9e21c5f5e59fccdf926e2b864232f308dcc0caacc80 && docker create --name profile...)  
└─ docker.io/roberthouse224/meditations-profile@sha256:b6dfc8b5758a9d92a9f5c9e21c5f5e59fccdf926e2b864232f308dcc0caacc80: Pulling from roberthouse224/meditations-profile  
Digest: sha256:b6dfc8b5758a9d92a9f5c9e21c5f5e59fccdf926e2b864232f308dcc0caacc80  
... +4 lines (ctrl+o to expand)
- Read 1 file (ctrl+o to expand)
- The profile requires one value from you:  
**GUTENBERG\_BASE\_URL** – the base URL of your Gutenberg mirror (e.g. http://localhost:8080).  
Please provide the URL. Note: if it's a localhost URL, Docker containers can't reach the host via localhost – use http://host.docker.internal:<port> instead.

http://host.█

▶ 0:00 / 2:04

→ INSERT → ▶ bypass permissions on (shift+tab to cycle)



# Issues

- No good way to "load" the servers
- No lifecycle management — no clean way to uninstall or reload servers
- `.mcp.json` wrangling — reading, merging, and writing config by hand is fragile



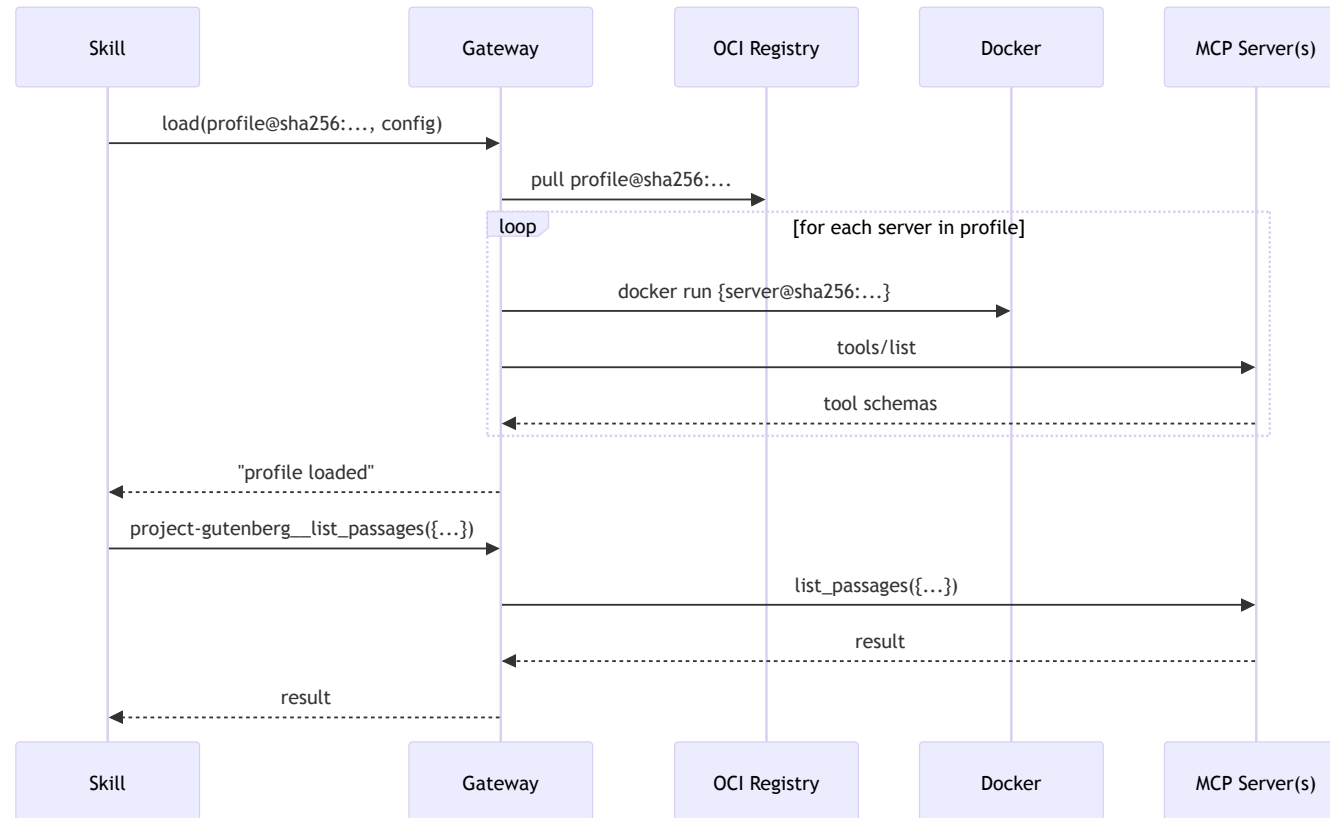
## Gateway + Profile Pattern

A thin, client-agnostic MCP layer that manages the runtime lifecycle of your servers.

- Reads the profile, pulls and loads servers
- Handles what the profile alone cannot — load, reload, unload
- Opens the door to further leveraging the dependency file



# Running the Skill




\* Claude Code

\* Claude Code 1 task start:mirror 2 +

mcp-dev-summit-2026 git:(mcp-dev-summit-2026) claude

Claude Code v2.1.84

Welcome back!



Sonnet 4.6 · API Usage Billing  
~/git/udsrvbtest/talks/mcp-dev-summit-2026

Tips for getting started

Run /init to create a CLAUDE.md file with instructions for Claude

Recent activity

No recent activity

/demos:meditations

- Before I can fetch a passage, I need two values from you:
  1. GUTENBERG\_BASE\_URL – the base URL of your Gutenberg mirror (e.g. <https://www.gutenberg.org>). If using a local mirror, use <http://host.docker.internal:<port>>.
  2. LOG\_DIR – an absolute path where the log file should be stored (e.g. `~/claude/meditations-log`). The directory will be created if it doesn't exist.

> GUTENBERG\_BASE\_URL: `http://host.docker.internal:8080` LOG\_DIR: `/Users/bobby/log`

-- INSERT -- ctrl+r to search history

▶ 0:00 / 0:48 🔊 🗄️ ⋮

16

# Closing

Package your MCP server dependencies as an OCI artifact:

- **Addressable** — a single immutable digest, no drift
- **Configurable** — placeholders for values the skill author or user supplies
- **Decoupled** — servers version and release independently

**Bonus:** pair with a gateway for lifecycle management — load, reload, unload without touching `.mcp.json`

*This is essentially what we built at Docker in [MCP Gateway](#) with the Profiles feature.*



## References



[github.com/bobbyhouse/talks](https://github.com/bobbyhouse/talks)



