

AI CATALOG

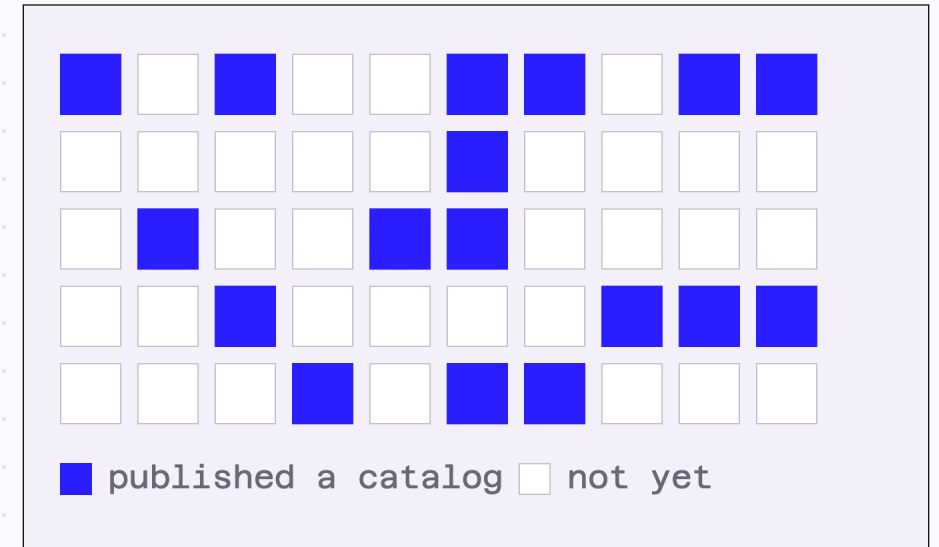
Decentralized discovery

mcp, skills, and beyond with ai catalog.

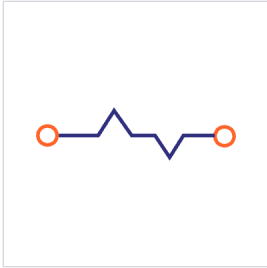
Tadas Antanavicius

warp.dev · mcp maintainer · pulsemcp

ai-catalog.io



Who Am I

<div>01</div> <div></div> <div>Co-Creator, MCP Registry</div> <div>mcp maintainer · community moderator</div>	<div>02</div> <div></div> <div>Co-Creator, PulseMCP</div> <div>one of the original MCP directories</div>	<div>03</div> <div></div> <div>Warp.dev</div> <div>fde · enterprise software factory infra</div>
---	---	--

NEW

WHY I CARE

MCP = an **open ecosystem** with no
gatekeepers, no native integrations

decentralized discovery is the next step in that journey

WHO CAME TOGETHER FOR THE AI CATALOG STANDARD

Buy-in across major standards and companies

01 standards	 mcp	 a2a	 skills	 plugins
02 steering committee	 Darrel Microsoft	 Junjie Google	 David Anthropic	 Tadas Warp.dev
03 contributors	<div></div>			<div> </div>

PART 01

MCP “discovery” today

mcp registries · plugin marketplaces · manual installation flows

THE PROBLEM

Vanilla agents are of limited use

a model with no tools can talk about your work. it can't touch it.

```
$ ship the jira release notes for v2.3
```

```
>_ i don't have access to jira. paste the tickets and  
i'll draft the notes.
```

```
$ then open a pr on github.com/acme/app
```

```
>_ i can't open pull requests. here is a git command you  
can run.
```

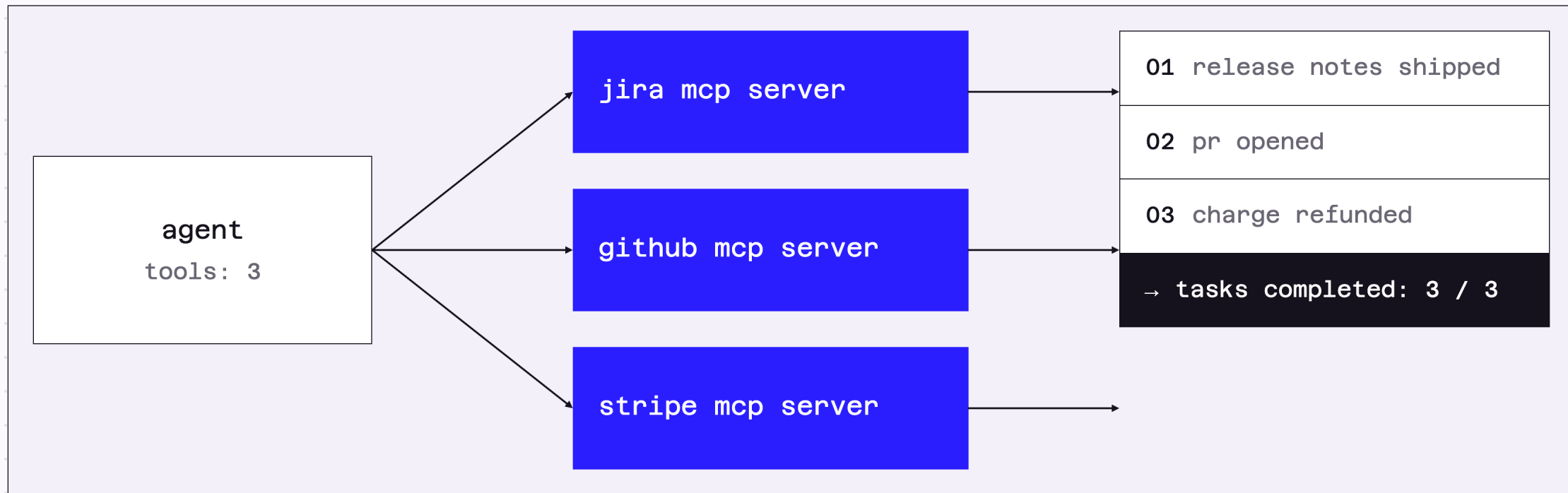
```
$ refund the stripe charge for order 8812
```

```
>_ i can't reach stripe. you'll need to do that in  
the dashboard.
```

```
tools: 0   tasks completed: 0 / 3
```

THE FIX

Users install MCP servers to unlock capabilities



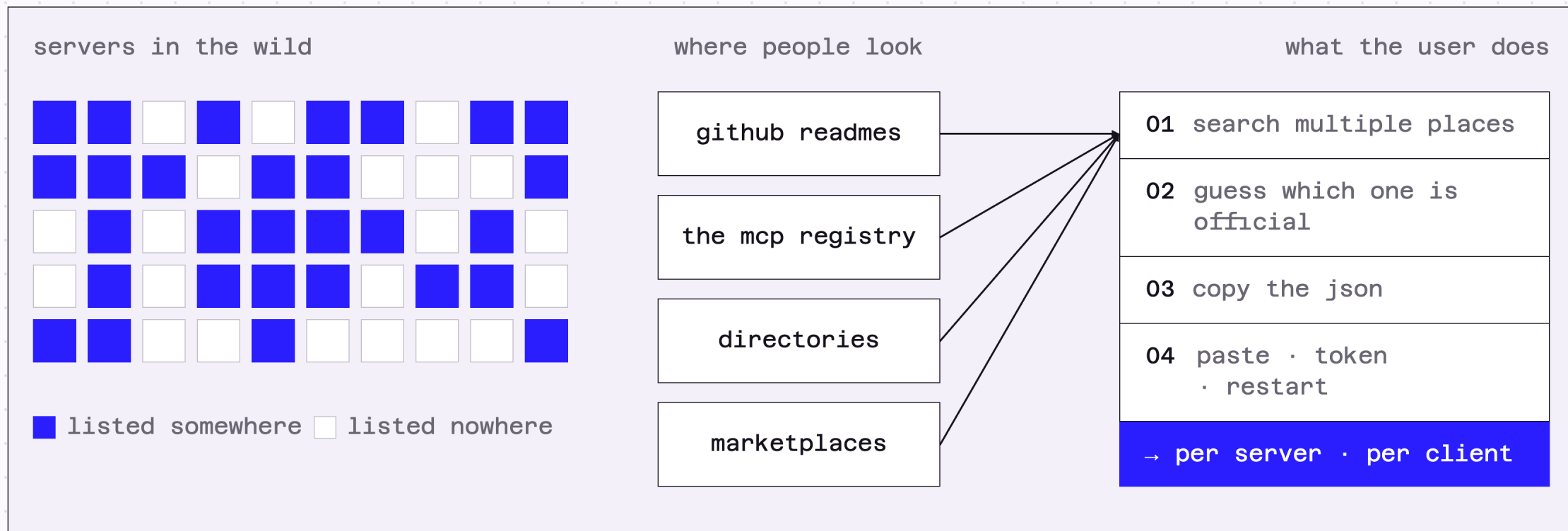
THE CATCH

But MCP installation is manual, unstandardized

GitHub READMEs a json snippet to copy, if the readme is current	Claude Marketplaces one client's shelf, one client's format	OpenAI Plugins another shelf, a slightly different list	Google Searches "jira mcp server" → which of these is official?
01 colleague tells you about a good server	02 hunt for it on google	03 wrangle the installation json	04 restart · repeat

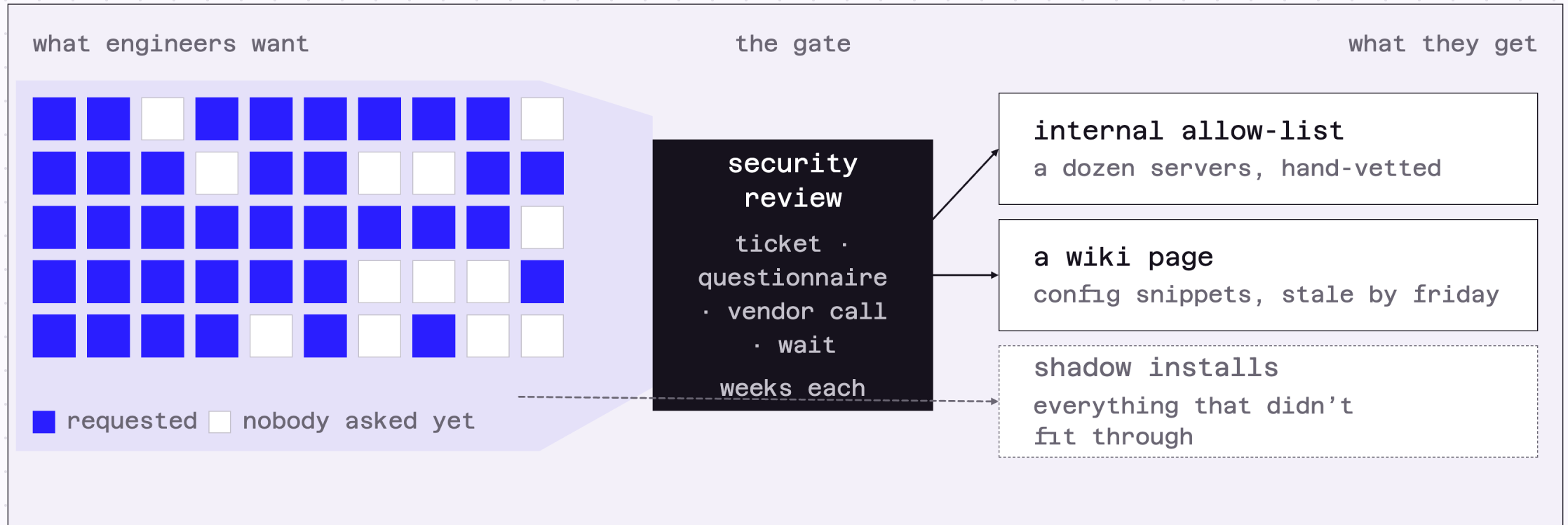
IN PUBLIC

Both in public, open-internet agents...



INSIDE THE FIREWALL

...and inside enterprises



THE COST

Who this hurts most

<p>01</p> <p>end users</p> <p>expanding capabilities is arduous</p> <hr/>	<p>02</p> <p>service providers</p> <p>discovery of your service is manual, user by user</p> <hr/>	<p>03</p> <p>enterprise IT teams</p> <p>vetting each server, one by one, version by version</p> <hr/>
--	--	--

PART 02

The AI Catalog solution

`/.well-known/ai-catalog.json`

AI CATALOG

A standard to publish your AI artifacts

one json file on your domain, listing everything you offer to agents for structured discovery

mcp

a2a

agent skills

agent plugins

...

ai-catalog.io

```
0  acme.com/.well-known/ai-catalog.json
1  {
0    "specVersion": "1.0",
2    "host": { "displayName": "Acme" },
0    "entries": [
3      { "type": "application/mcp-server-card+json",
0    ... },
4      { "type": "application/a2a-agent-card+json",
0    ... },
5      { "type": "application/agent-skills+zip", ... },
0      { "type": "application/agent-plugin+json", ... }
6    ]
0  }
7
0
8
0
9
1
0
1
1
1
1
2
```

Standards maintain their own entry format

```
the catalog entry · ai catalog's job

{
  "type":
    "application/mcp-server-card+json",
  "url":
    "https://example.com/mcp/server-card",
  "trustManifest": { ... }
}
```

what it is · where it lives · who vouches.
nothing about how to connect.



```
🔗 the artifact · mcp's job: a server card

{
  "$schema": ".../v1/server-card.schema.json",
  "name": "example-org/with-remote",
  "version": "2.1.0",
  "remotes": [{
    "type": "streamable-http",
    "url": "https://{tenant}.example.com/mcp",
    "headers": [{
      "name": "Authorization",
      "value": "Bearer {token}", "isSecret":
true
    }]
  }]
}
```

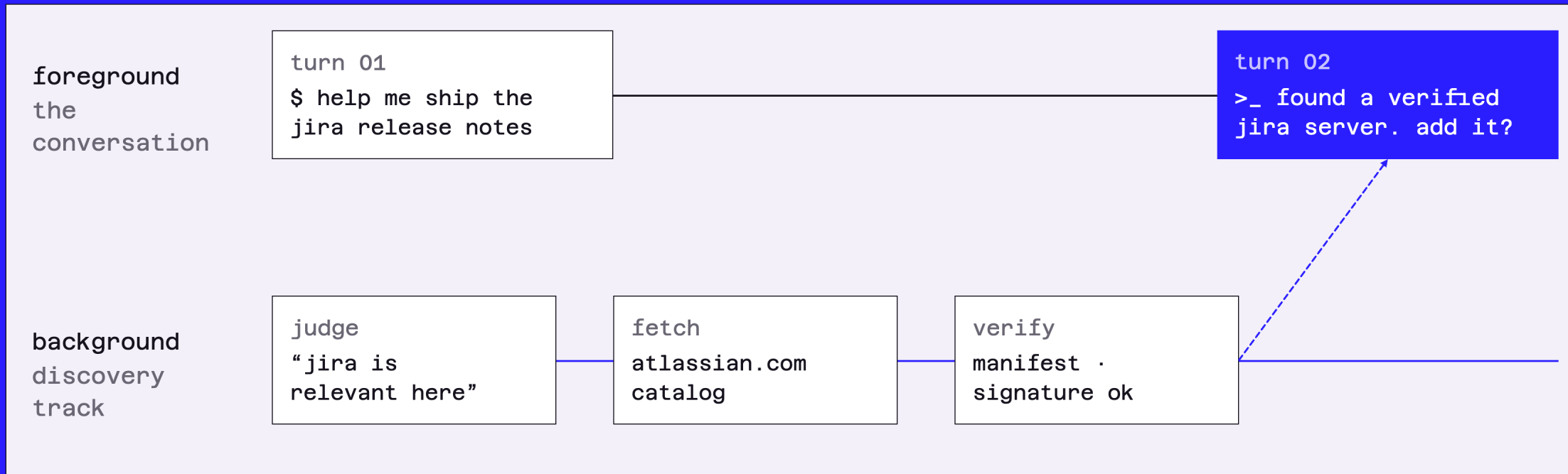
mcp server cards: github.com/modelcontextprotocol/experimental-ext-server-card

THE CATCH, SOLVED

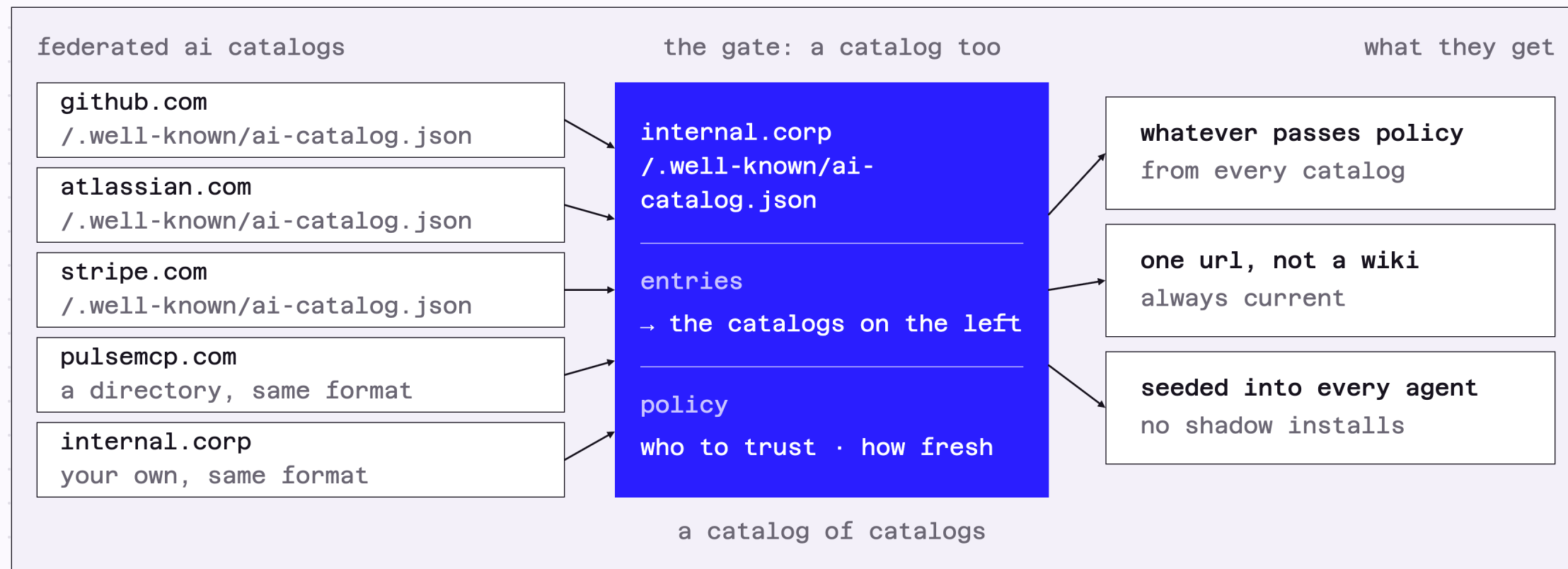
Installation becomes standardized, programmatic

<code>client a</code> <code>GET atlassian.com</code> <code>/.well-known</code> <code>/ai-catalog.json</code> <code>→ jira · installed</code>	<code>client b</code> <code>GET atlassian.com</code> <code>/.well-known</code> <code>/ai-catalog.json</code> <code>→ jira · installed</code>	<code>client c</code> <code>GET atlassian.com</code> <code>/.well-known</code> <code>/ai-catalog.json</code> <code>→ jira · installed</code>	<code>client d</code> <code>GET atlassian.com</code> <code>/.well-known</code> <code>/ai-catalog.json</code> <code>→ jira · installed</code>
<code>01</code> <code>publisher writes</code> <code>one file</code>	<code>02</code> <code>any client fetches</code> <code>it mid-session</code>	<code>03</code> <code>install on-the-fly</code>	<code>04</code> <code>no restart · no UI</code> <code>spelunking</code>

Dual-track background discovery



...and inside enterprises: the gate points to a federation of AI Catalogs



HOW THE GATE VERIFIES

The trustManifest

identifier

type

url

version

trustManifest →

a signed sticky note on an entry: who vouches, for which bytes, with what evidence, until when.

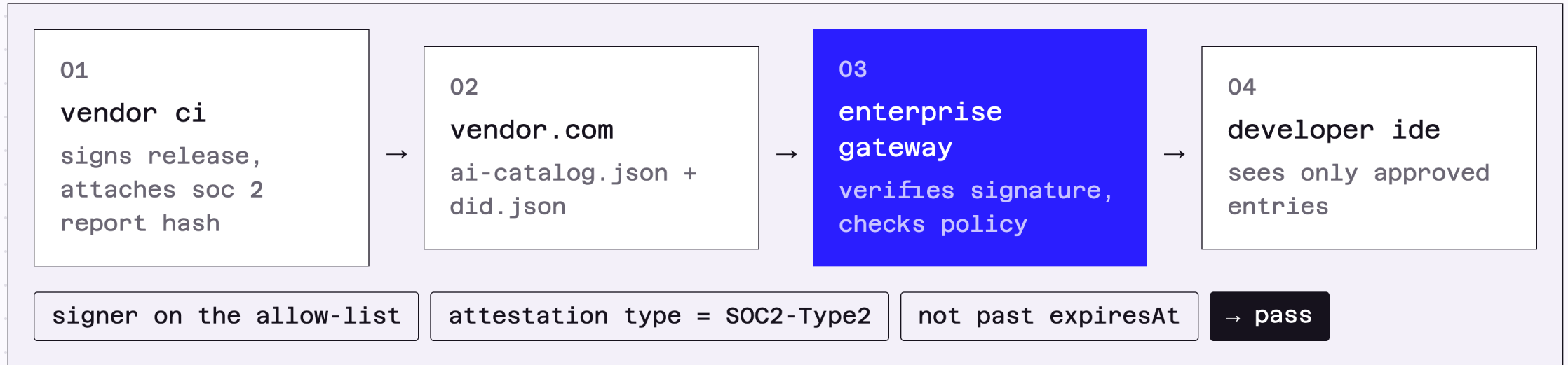
01	WHO	identity · did:web:acme.com
02	WHAT	subject · exact bytes, pinned by hash
03	EVIDENCE	attestations · provenance
04	WHEN	issuedAt · expiresAt
05	PROOF	signature · es256 over 01-04

go deeper
1:45pm
1120c ·
lower
level

Interoperable Agent Discovery: AI Catalog, ARD, and the AGNTCY Directory

Luca Muscariello, Cisco · Junjie Bu, Google

“Only SOC 2 compliant servers reach our engineers”

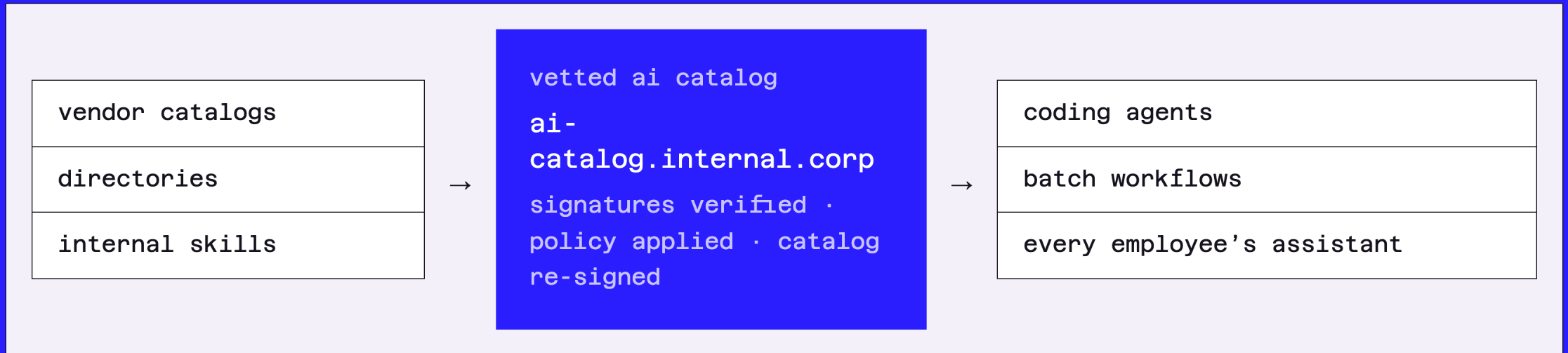


A second voice on the same release



neither party can alter the other's manifest.
consumers choose whose voices count.

A vetted, federated catalog seeds the whole system



```
catalogs:  
  - url: https://ai-catalog.internal.corp  
    trust: did:web:corp.com
```

```
mcp servers: 14 seeded  
skills: 31 seeded  
per-agent setup: 0
```

PART 03

What's ahead for AI Catalog

`publishers · catalogs · contributors · consumers`

The base case: one entity does it all

01 · 02 · 03

publisher = catalog = contributor

e.g. [github](#), on [github.com](#)

owns the domain, hosts its own catalog, vouches for its own entries. the only thing you verify is that it is who it says it is.

04

consumer

e.g. [warp harness](#)

the only other party.
runs the checks, applies
its own policy.

and this might be enough for the open-internet use case

Split it once: someone else vets what the publisher ships

<p>01</p> <p>publisher</p> <p>e.g. github</p> <p>owns the domain in the urn:air. still the only one who can sign for that entry.</p>	<p>02 · 03</p> <p>catalog + contributor</p> <p>e.g. internally managed enterprise catalog</p> <p>hosts a catalog of other people's entries and adds signed claims about them. the enterprise trusts it to do the vetting.</p>	<p>04</p> <p>consumer</p> <p>e.g. warp harness</p> <p>verifies the publisher's signature and internal claims, then applies policy.</p>
---	--	---

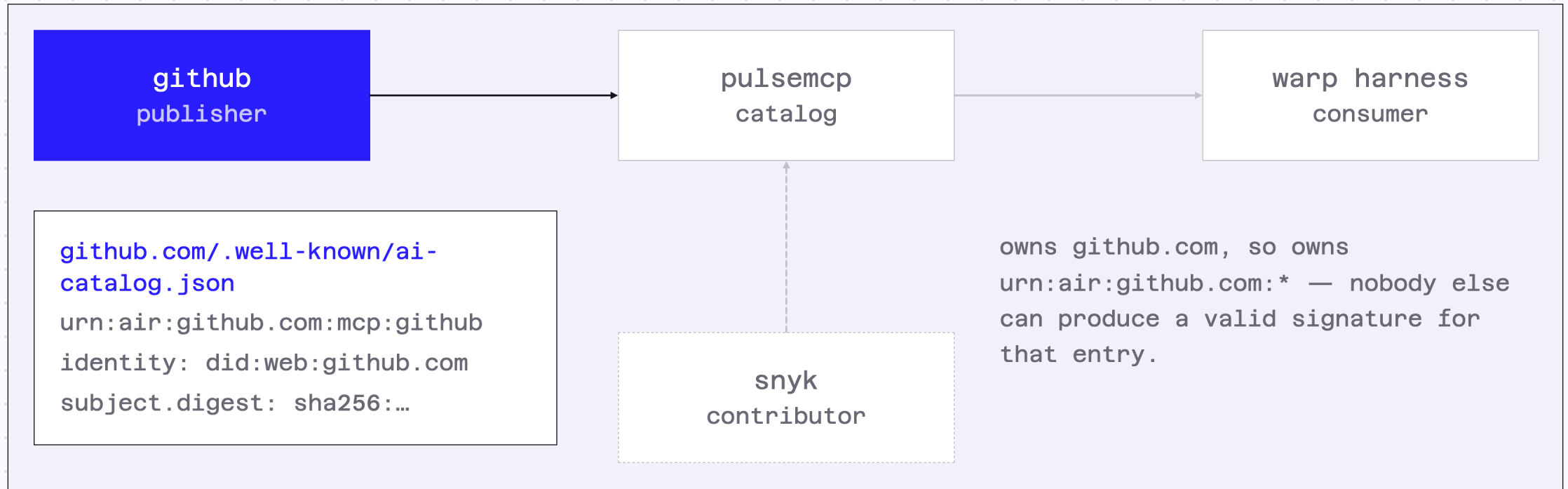
the publisher never has to know who vetted it: great for the enterprise use case

End-state: all four are different companies

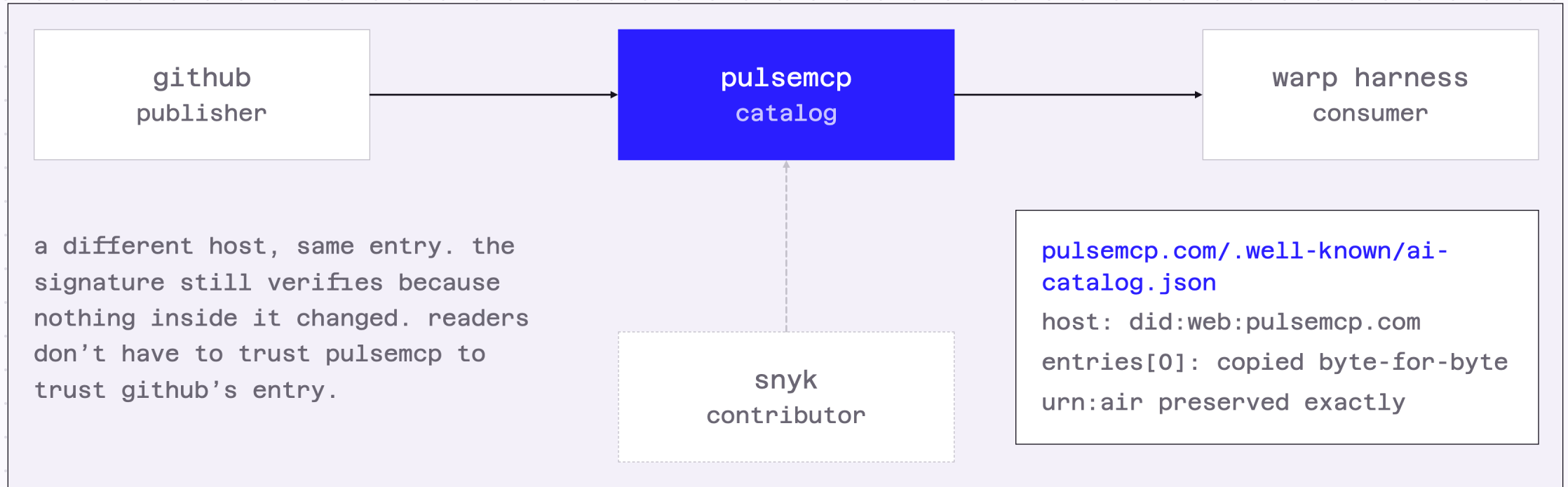
<p>01</p> <p>publisher</p> <p>e.g. github</p> <p>owns the domain in the urn:air. only it can sign for that entry.</p>	<p>02</p> <p>catalog</p> <p>e.g. pulsemcp</p> <p>hosts a catalog file. you don't have to trust it to trust the entry.</p>	<p>03</p> <p>contributor</p> <p>e.g. snyk</p> <p>adds a signed claim about someone else's artifact.</p>	<p>04</p> <p>consumer</p> <p>e.g. warp harness</p> <p>runs the checks, then applies its own policy.</p>
--	--	--	--

catalogs and contributors can serve a rich enterprise AI artifact discovery and management ecosystem

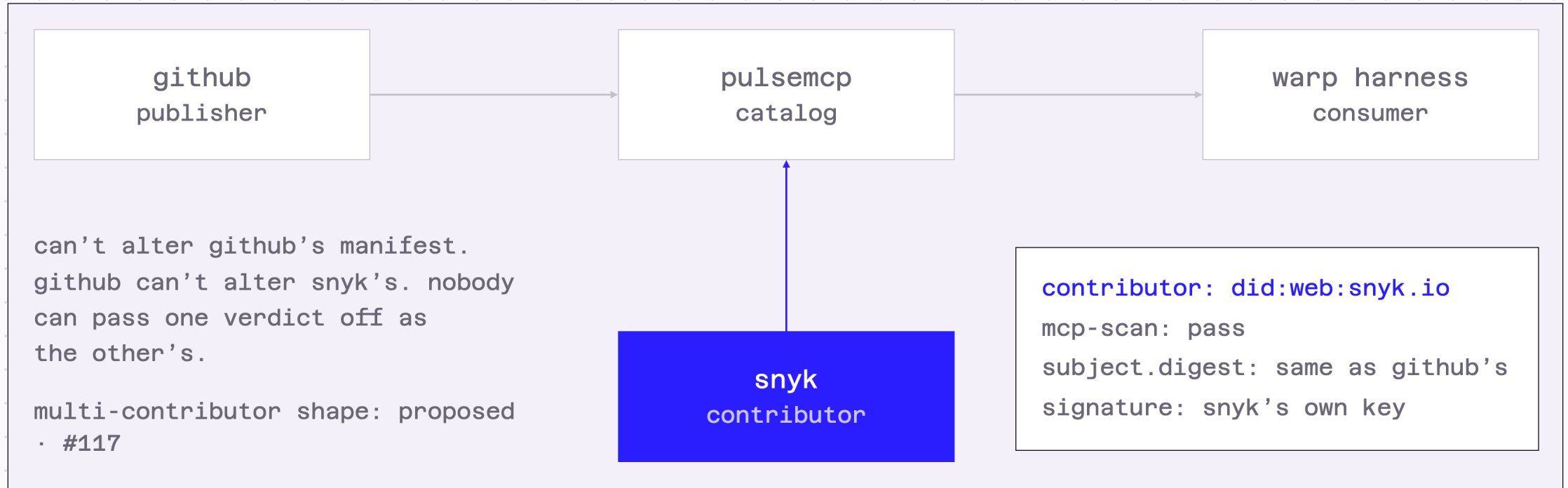
GitHub signs its own MCP server



PulseMCP relists it, signature intact

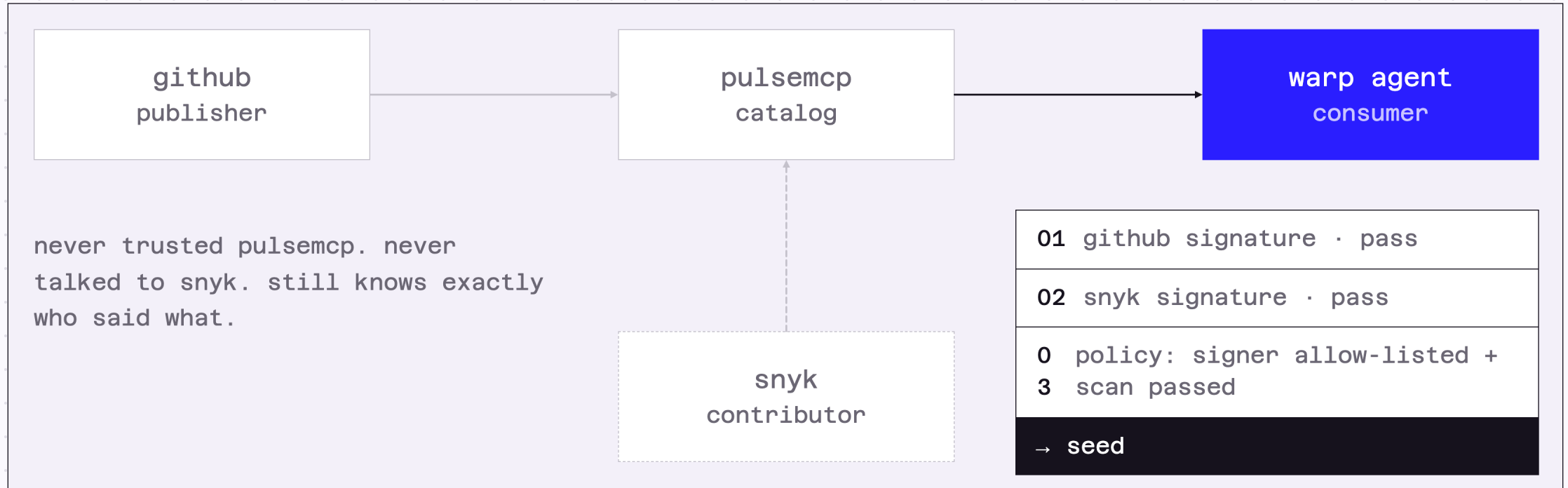


Snyk adds its own signed verdict



CONSUMER

The Warp agent verifies, then decides



Thank you

01 · join us on discord

bit.ly/ai-catalog-discord

anyone welcome



02 · warp.dev

Open infrastructure for enterprises to build software factories

come talk to me, tadas@warp.dev

03 · go

deeper

1:45pm ·

1120c

Interoperable Agent Discovery: AI Catalog, ARD, and the AGNTCY Directory

Luca Muscariello, Cisco · Junjie Bu, Google

warp.dev · x.com/tadasayy

ai-catalog.io