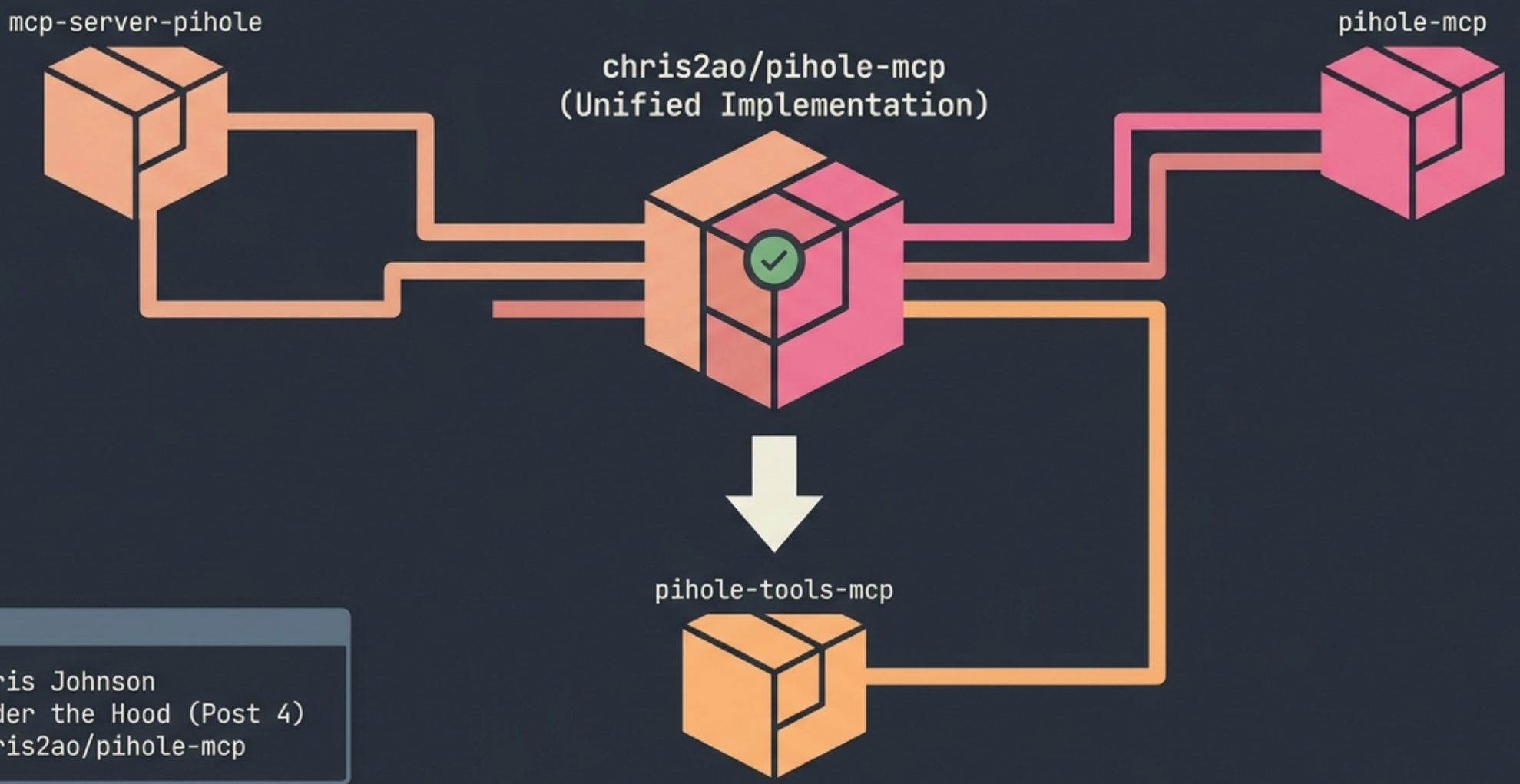
Consolidating the Pi-hole MCP Ecosystem

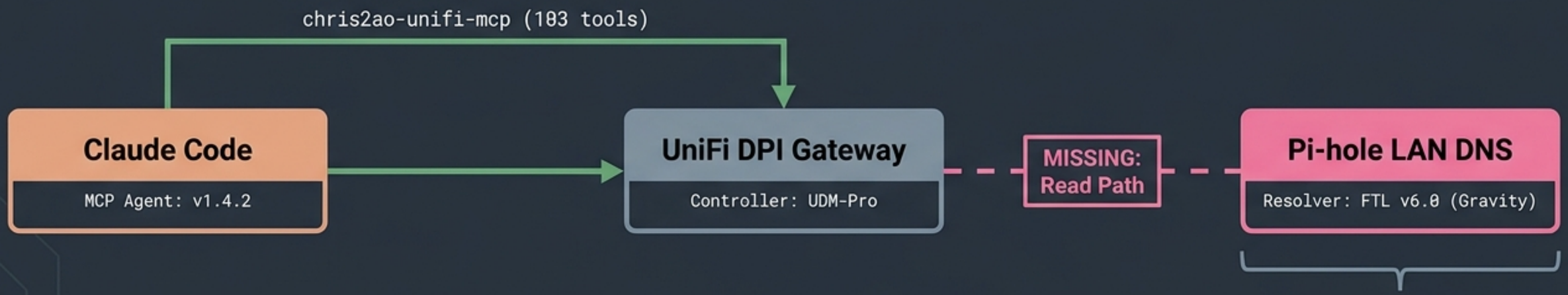
Synthesizing 3 upstream repositories into a single, battle-tested Python implementation.



Author: Chris Johnson
Series: Under the Hood (Post 4)
Target: chris2ao/pihole-mcp

The Blind Spot in the AI's Network Visibility

The home lab already had Claude integrated with the **UniFi** gateway via a custom MCP. However, the AI had zero visibility into local DNS queries and ad-blocking rules handled by Pi-hole.

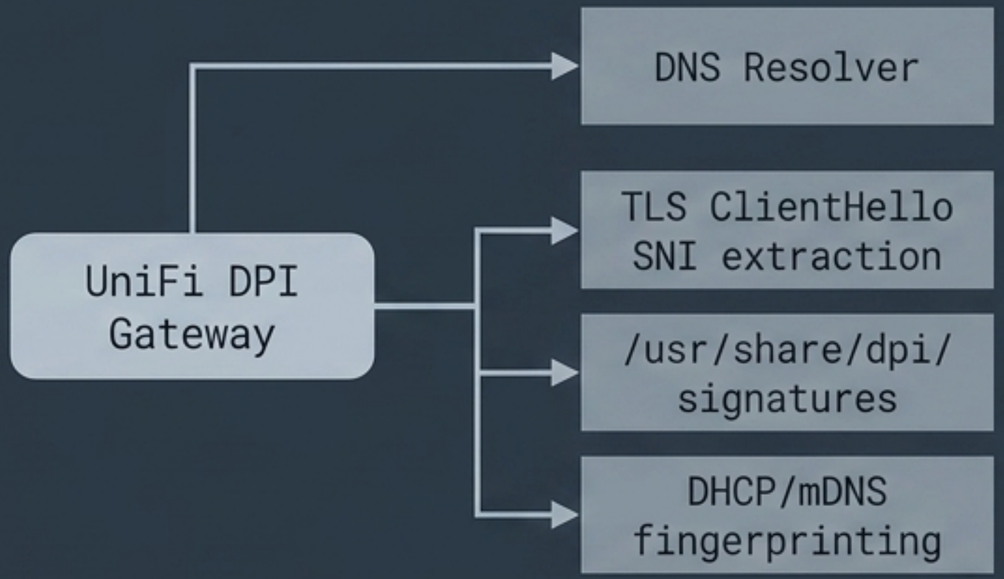


- Handles every client DNS query
- Returns null for blocklist (gravity) domains

Validating Assumptions with Pre-Build AI Research

Hardware Assumption

Does Pi-hole as LAN DNS degrade UniFi DPI?



Result: No degradation.

Software Reality

What Pi-hole MCPs already exist?

```
> /deep-research existing pi-hole mcp servers

Search complete.
5 existing servers found.
10 major capability gaps remain.
```

Editorial Developer Environment

Mapping a Fragmented and Stale Ecosystem

Repository	Language Stack	Tools	Auth Support	Status (2026)
aplaceforallmystuff	TypeScript	14	App Password (v6)	Active
sbarbett	Python		in Pass (v6)	Active
brettbergin	Python	6	API + Pass (v5/v6)	Stale
cwdcwd	TypeScript	16	API + Pass (v5/v6)	Stale
sebszczec	Python	4	App Password (v6)	Stale

Language stack mismatch.

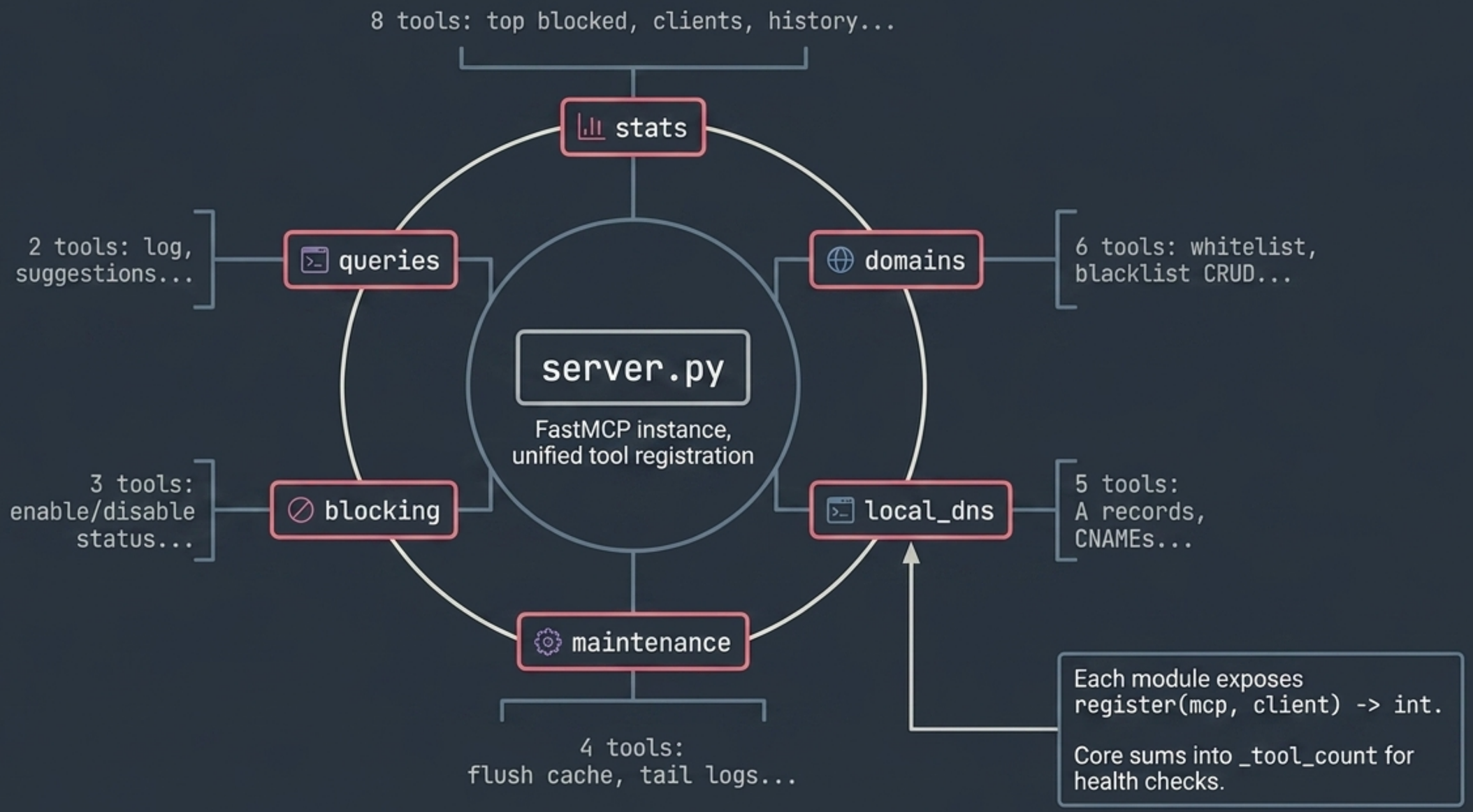
10 critical capability gaps remain (adlist CRUD, Teleporter, VLAN analytics).

Toolchain Uniformity Beats Best-of-Breed for Second Integrations

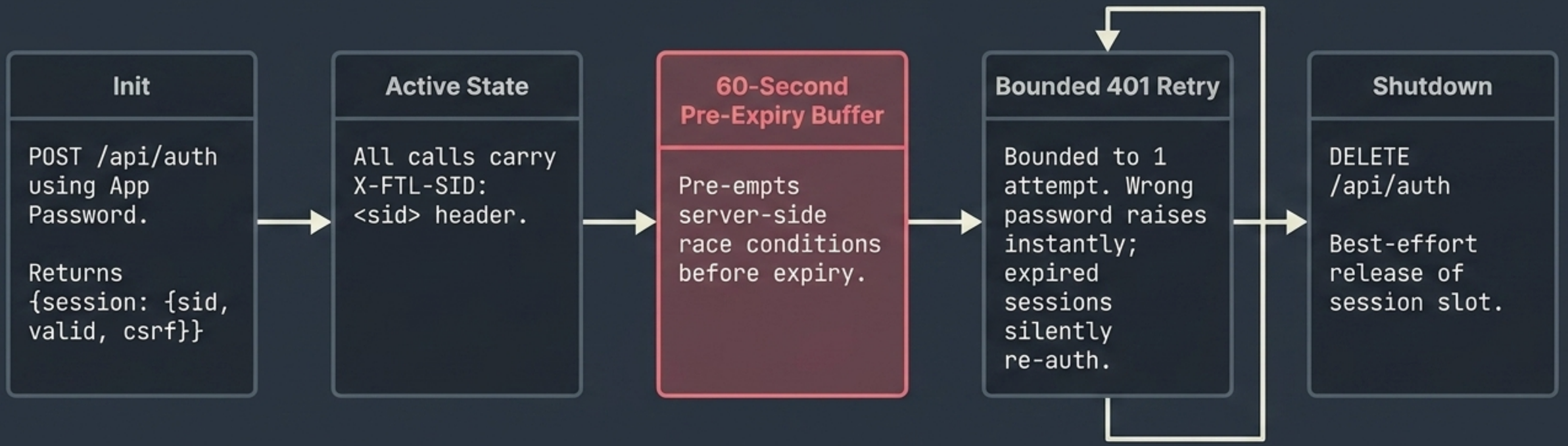
The useful tool surface was split across three different repos. Consolidating the best ideas into a new repo matched the author's existing UniFi MCP stack, maximizing long-term maintainability.


```
<Python> | <FastMCP> | <httpx> | <Pydantic v2> | <pytest> | <respx> | <uv> | <hatchling>
```

System Architecture: 28 Tools Across 6 Modules



Designing a Resilient Session Auth Pattern



Navigating the Local DNS Read-Modify-Write Trap

READ

Fetch hosts string array from `/api/config/dns`.

```
["192.168.x.x router"]
```

WRITE

`PATCH` the entire array back to the server.

MODIFY

Shallow merge operation locally. Strip existing entries for target host, append new one.

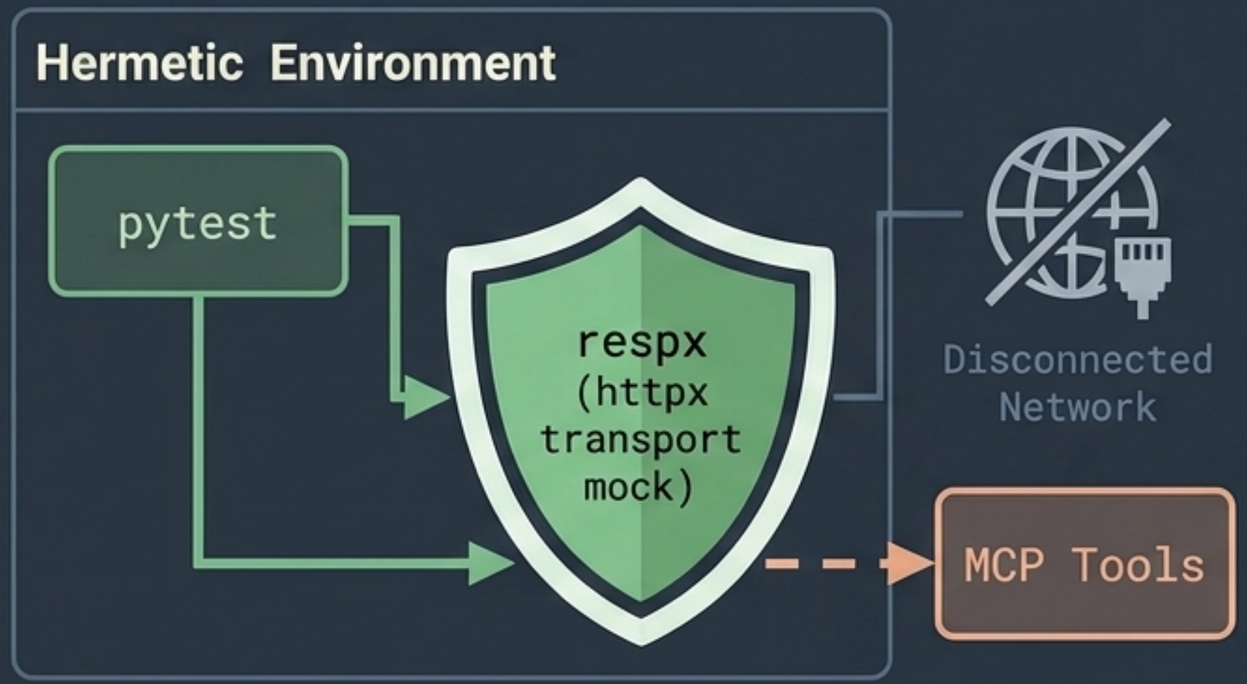
Creates a single-user race condition. Acceptable trade-off for isolated home lab environments.

Proving Reliability with Hermetic Testing

13 Passing Tests
TOTAL SUITE

0.48s Execution Time
MEDIAN RUNTIME

Python 3.12 & 3.13 CI
SUPPORTED RUNTIMES

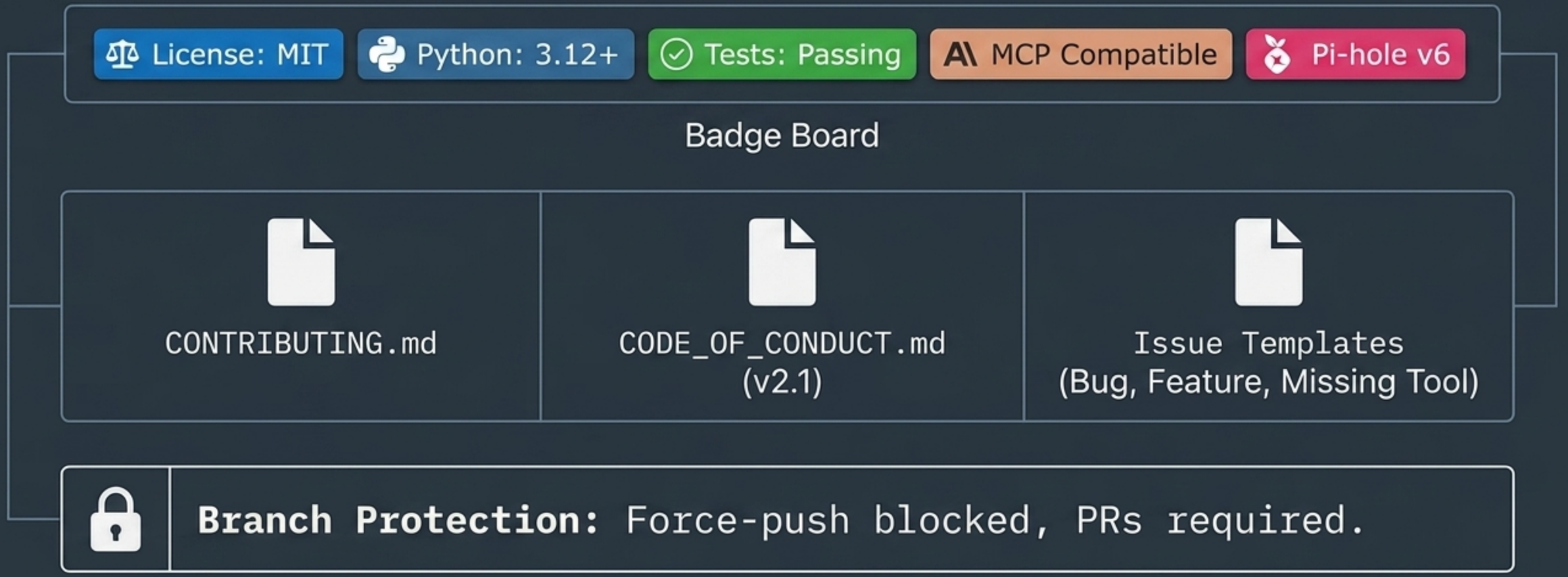


Test Suite Breakdown

- test_config.py: URL normalization, /admin suffix stripping.
- test_client.py: Auth success/failure, session caching, 401 retry loops.
- test_tools.py: Tool registration validation (28 sum), local DNS delta patches.

Establishing the Open-Source Polish Standard

High-quality hygiene is reusable. The CI matrices, templates, and branch protections designed for this consolidation were immediately retrofitted to the older UniFi MCP project. **Ship once, apply everywhere.**



Architectural Framework: MCP Tools vs. CLI Skills

Use MCP

Profile: Read-only, low-risk, observational.

Invocation: Model discovers and composes it autonomously.

Audit: Traceable at the tool-call level.

Use Skill

Profile: Destructive, state-changing actions.

Invocation: Explicitly triggered via slash-command only. Never autonomous.

Audit: Requires pre-action snapshots and preview-then-apply workflows.

Future Roadmap and Abstracted Engineering Lessons

Future Roadmap

```
❯
[ ] Adlist CRUD (/api/lists)
[ ] Teleporter backup
[ ] Per-VLAN analytics via CIDR
[ ] /pihole-* skills family for destructive ops
```

Extracted Philosophies

1. Run /deep-research before non-trivial builds to radically shape architecture.

2. Toolchain uniformity strictly beats best-of-breed for secondary integrations.

3. Extract complex behaviors (like session auth and 401 retries) into reusable, bounded shapes.

Repository public at github.com/chris2ao/pihole-mcp. Relaæd under MIT.