



Agent AI Foundation

AGNTCon + MCPCon  
North America

# Generation–Verification Asymmetry

The unnamed failure pattern behind production incidents

**Birajendu Sahu** Senior Engineering Manager, Endor Labs





Agent AI Foundation

AGNTCon + MCPCon  
North America

# Everything is green.

 feat: load service config from YAML Opened by pr-agent · 4 turns, all agent-authored

✓ CI: pytest, ruff, mypy passed

✓ Review agent: VERDICT: APPROVE

✓ Ready to merge



Every turn done by  
agents.



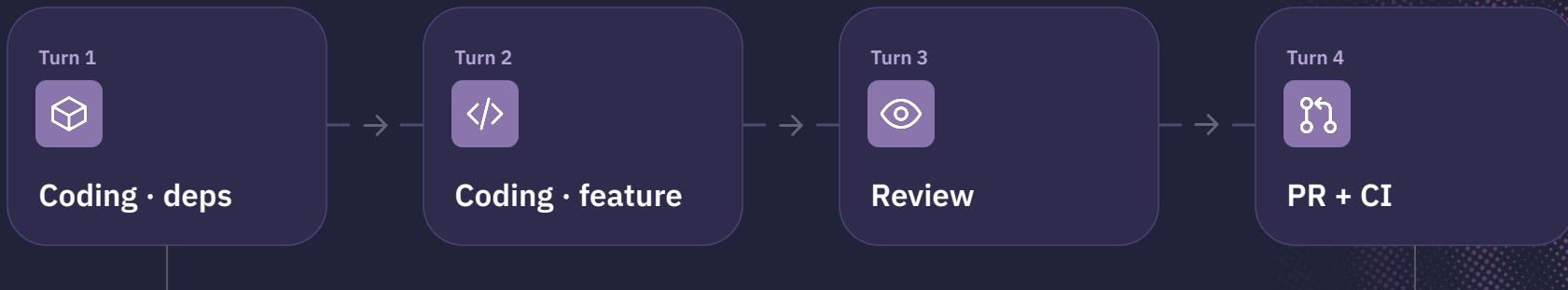
Every check passed.



**It ships an RCE-class  
advisory.**

# Four turns. No verification.

Watch it ship.



Runs in a real environment



Real sessions



Real git worktree



Real CI exit code



Least privilege per turn

# The risk isn't in the diff

It was in how the diff met code that already existed.

Turn 1 · the only line changed

```
+ pyyaml==5.3.1
```

Already in the repo · legacy\_config.py:69

```
yaml.load(text,  
  Loader=yaml.FullLoader)
```

Turns 3 and 4

✓ Reviewer: APPROVE

✓ CI: green

Reachable call path

POST /admin/config



handlers.parse\_update("legacy")



legacy\_config.loads()

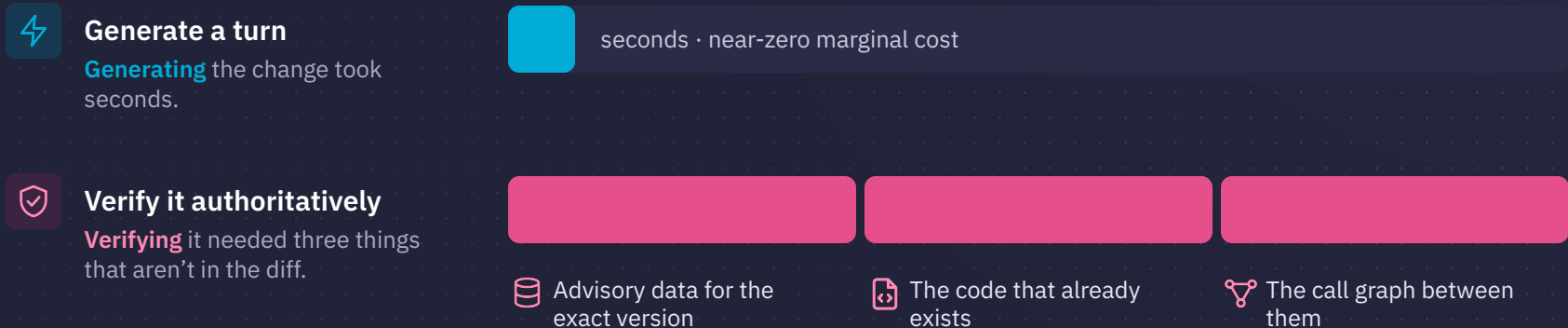


⚠ `yaml.load(FullLoader)`

CVE-2020-14343 · arbitrary code execution

# Generation-verification asymmetry

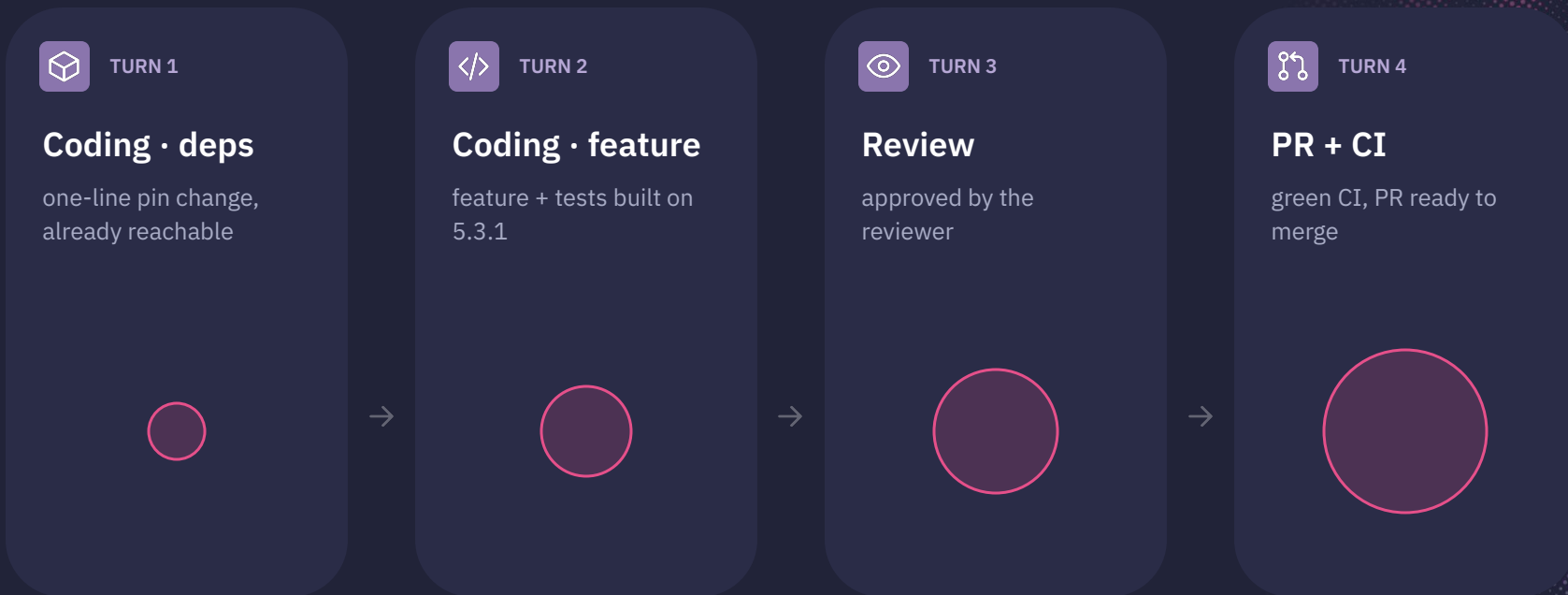
Every turn you don't verify is a foundation the next turn builds on.



Illustrative, not to scale

# Unverified turns compound

Same check at every turn. **Growing cost of skipping it.**



Where the **agentic stack** is **being built**.

# This is not hypothetical

The gap isn't model quality. **It's what the agent can check.**



**19.6%**

of packages referenced in  
LLM-generated code did not exist

Spracklen et al., USENIX Security '25 (576k  
samples, 16 models)



**49%**

of dependency versions imported  
by AI coding agents had known  
vulnerabilities

Endor Labs, State of Dependency Management  
2025



**3x**

safer dependency choices when  
agents were given validation  
tooling

Endor Labs, State of Dependency Management 2025

# Verification that isn't



## Another LLM reviewer

👁 LOOKS LIKE

A second agent reads the diff.

❗ FAILS BECAUSE

The risk isn't in the diff. No advisory data, no call graph.



## Confidence scores

👁 LOOKS LIKE

"Risk: low (0.82)"

❗ FAILS BECAUSE

A number with no evidence behind it can't be audited or acted on.



## Retry until pass

👁 LOOKS LIKE

The agent loops until the check goes green.

❗ FAILS BECAUSE

Optimizes for the check, not the property. Agents learn to route around it.

# Same reviewer. Asked only about security

⚠️ **Fluent. Confident. Unverifiable.**



**review-agent:**

"Dependency versions now match billing-service. No security concerns in this change. VERDICT: APPROVE"

# Verification-first design



## Ground truth, not opinion

01

Advisory data and a call graph. Never an LLM's judgment.



## Evidence the agent can act on

02

Severity, reachability, call path, remediation. Not just "a CVE exists".



## At the moment of change

03

Verify when the manifest is written, before the next turn builds on it.



## Enforced by the system, fails closed

04

Verifier down: closed. Reachability unknown: reachable. Two blocks: no PR.



Agentic AI Foundation

AGNTCon + MCPCon  
North America

# The evidence the gate returns

```
{  
  "decision": "block",  
  "package": "pyyaml",  
  "version": "5.3.1",  
  "advisory": "GHSA-8q59-q68h-6hv4",  
  "cve": "CVE-2020-14343",  
  "severity": "high",  
  "reachable": true,  
  "call_path": [  
    "POST /admin/config",  
    "handlers.parse_update",  
    "legacy_config.loads",  
    "yaml.load(FullLoader)"  
  ],  
  "remediation": {  
    "upgrade_to": "5.4"  
  },  
  "evidence": {  
    "backend": "endorlabs",  
    "checked_at": "2026-10-22T..."  
  }  
}
```



## decision

The write is not applied.



## reachable + call\_path

Why it matters in this code, not just that it exists.



## remediation

The agent's way forward: the lowest fixed version.



## evidence

Saved per gate check and added to the PR. Auditable.



Agentic AI Foundation

AGNTCon + MCPCon  
North America

# Verify at the moment of change

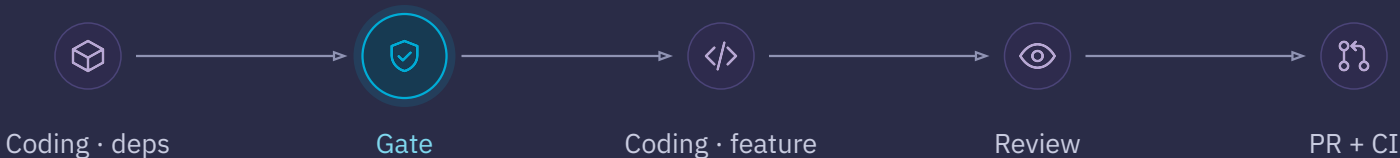
RUN 1 · AFTER THE FACT

The reviewer sees the diff. The risk isn't in the diff.



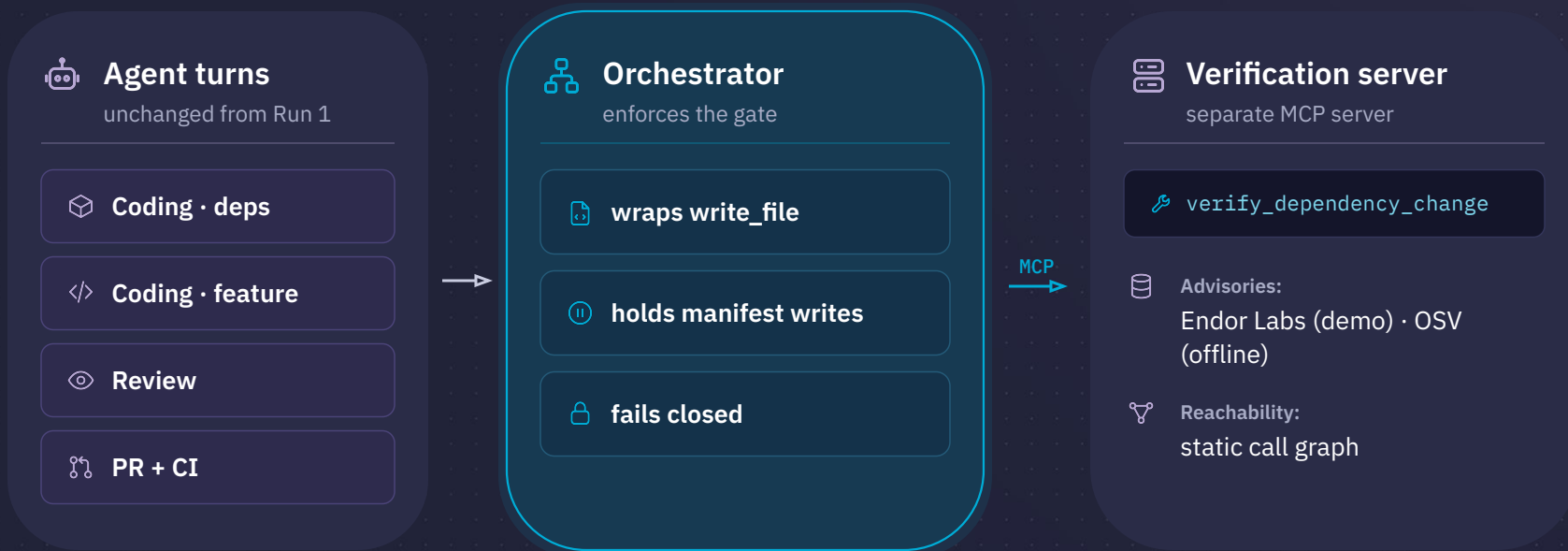
RUN 2 · AT THE MOMENT OF CHANGE

The write isn't applied until it's verified. The agent fixes it in the same turn.



Where the **agentic stack** is **being built**.

# How the gate works



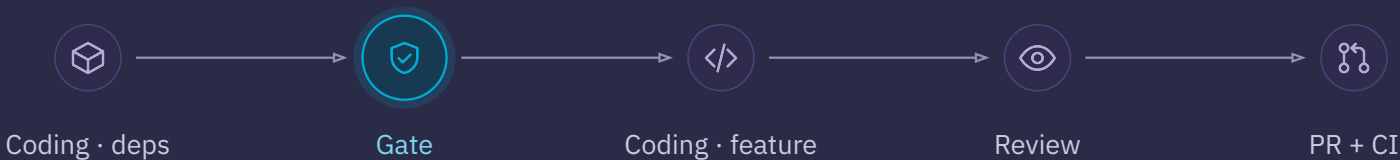
## POLICY

Block if reachable and severity  $\geq$  high. Unknown reachability counts as reachable.



# Same task. Same agents. Same model.

One gate on manifest writes.



# What changed: **one gate**



Outcome

⊗ Run 1

CI green, vulnerable

✓ Run 2

CI green, actually safe

|                      |                                   |                                        |
|----------------------|-----------------------------------|----------------------------------------|
| When it's verified   | Never (review sees only the diff) | When the change is written             |
| Who enforces it      | Nobody                            | The orchestrator, not a prompt         |
| What the agent gets  | Nothing                           | Advisory, reachability, call path, fix |
| What the agent does  | Ships the vulnerable pin          | Corrects itself in the same turn       |
| Unreachable advisory | n/a                               | Allowed, with the reason recorded      |
| Audit trail          | A PR that looks clean             | Gate records in commits and PR         |



Agentic AI Foundation

AGNTCon + MCPCon  
North America

# Not just dependencies

Ground truth, at the moment of change, with evidence.

WHAT THE AGENT WRITES

 Dependency manifests

 License choices

 Terraform / Helm changes

 Schema migrations

GATE WITH GROUND TRUTH

 Advisory data + reachability

 License policy engine

 Infrastructure policy as code

 Schema diff + data quality checks

01 Verify where the change happens, not at review time.

---

02 Evidence closes the gap, not another LLM reviewer.

---

03 Enforce it in the system, not in a prompt.

---

04 Good evidence turns a block into a fix.





Agentic AI Foundation

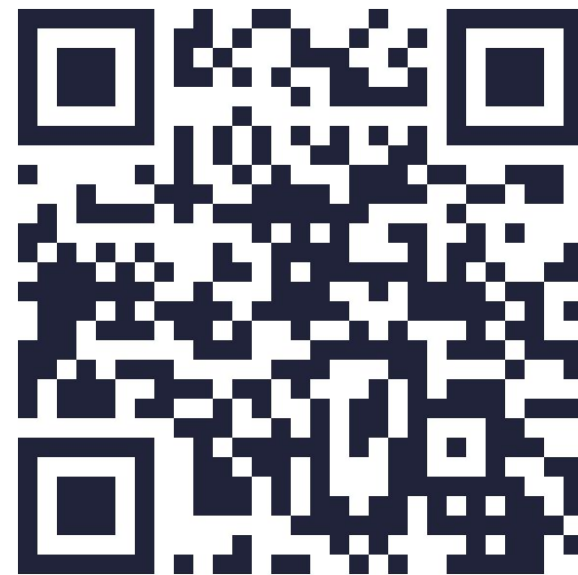
**AGNTCon + MCPCon**  
North America

# Thank you. Questions?

Birajendu Prasanna Sahu

Senior Engineering Manager

Endor labs



Lets connect: <https://www.linkedin.com/in/birajendup/>