Career Path

AI SECURITY ENGINEER

Securing AI applications from design to production

Follow one realistic case: Westbridge Mutual builds a claims-review AI assistant that must be useful, secure, and auditable.

ITLearn360 Original Career Series

Why AI Security Is Different

AI introduces new security boundaries, but it does not remove classic AppSec, IAM, cloud, and data-protection work.

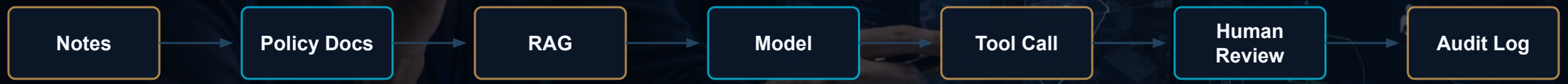
- The model is only one component in a larger system
- Security failures can emerge between prompts, data, tools, and users
- The job is about risk reduction, testing evidence, and secure design

CONTINUING SCENARIO

Westbridge Mutual: The System to Secure

An internal claims-review assistant can read adjuster notes, search approved policy documents, summarize evidence, and draft follow-up questions.

- Must not approve or deny claims
- Must not reveal another customer's data
- Must not follow hidden instructions inside documents
- Must not write to systems without authorization



ASSET VIEW

What the AI Security Engineer Protects

Customer data

Claims, PII, uploaded files

Model access

API keys, routes, quotas

Knowledge layer

Policies, vectors, metadata

Decision flow

Recommendations, approvals, audit

Tool permissions

Read/write actions and identities

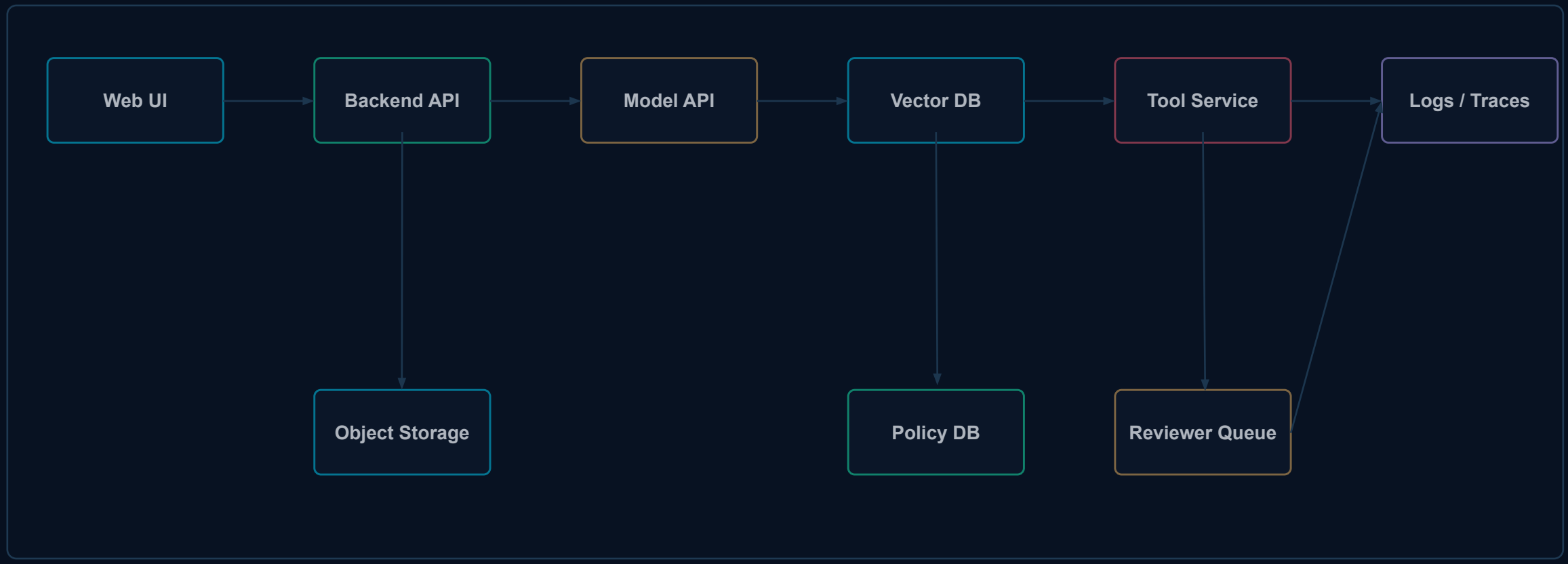
Evidence trail

Prompts, traces, tests, logs

Security starts by naming what would be damaged if the system behaves incorrectly.

TRUST-BOUNDARY MAP

Map the System Before Testing



Ask: Which data crosses which boundary, under whose identity, and with what permission?

DO NOT SKIP THE BASICS

Classic AppSec Still Applies

Traditional controls

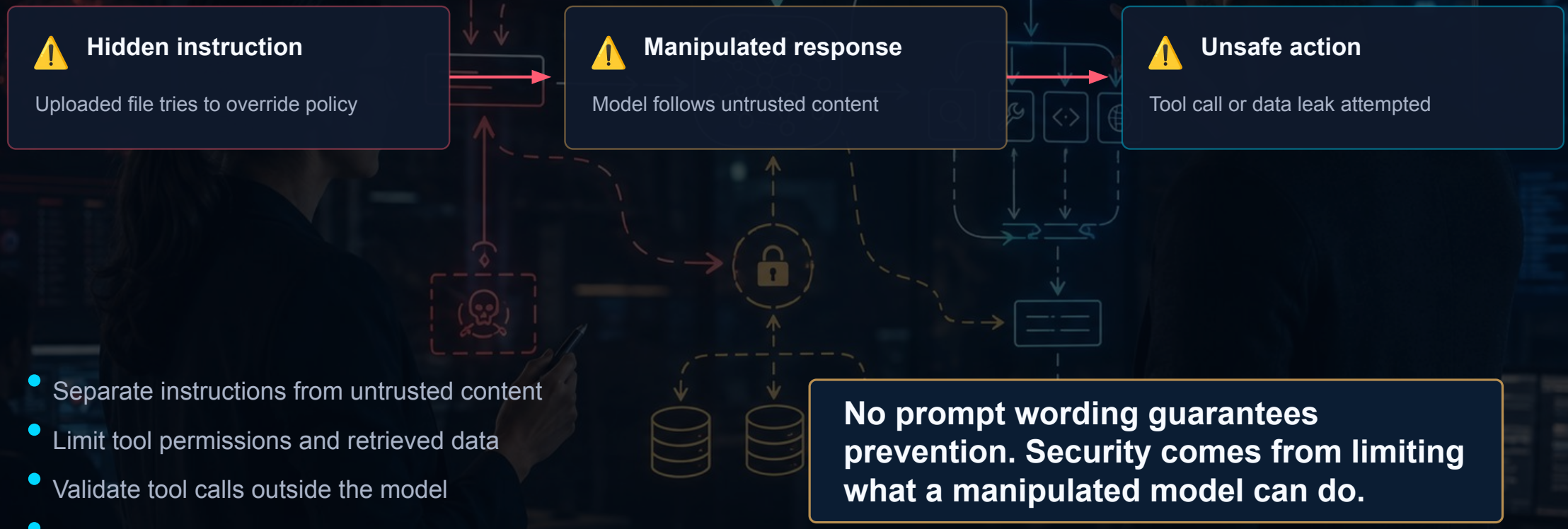
- AuthN/AuthZ
- Session security
- Input validation
- Output encoding
- API security

AI-specific additions

- Secrets management
- Dependency scanning
- Secure cloud config
- Logging & rate limits
- Vulnerability management

Generated output is untrusted input until deterministic application code validates it.

Prompt Injection: Reduce Impact

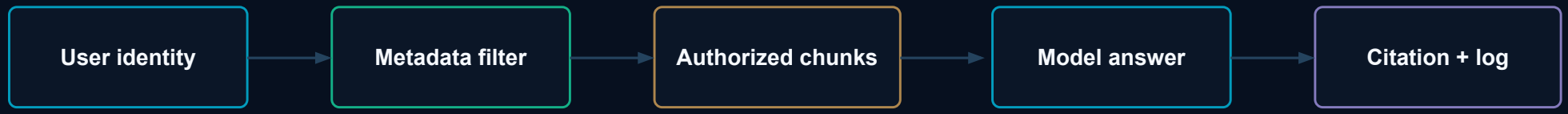


- Separate instructions from untrusted content
- Limit tool permissions and retrieved data
- Validate tool calls outside the model
- Require approval before consequential actions

No prompt wording guarantees prevention. Security comes from limiting what a manipulated model can do.

RETRIEVAL SECURITY

Secure RAG: Permissions Travel With Evidence



- ✓ Ownership
- ✓ Classification
- ✓ Source
- ✓ Version
- ✓ Effective date
- ✓ Access rules
- ✓ Retention
- ✓ Revocation

A vector database is not an authorization system. Access filtering happens before evidence reaches the model.

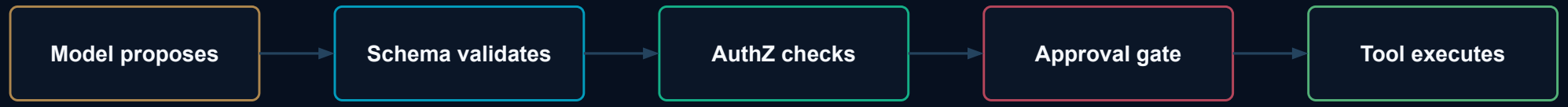
PROVENANCE

AI Supply Chain: Models Are Dependencies

Models	**Weights, hosted APIs, adapters**	**Verify** source • version • license • hash/signature • maintenance • approved use
Data	**Training, fine-tuning, evaluation**	**Verify** source • version • license • hash/signature • maintenance • approved use
Code	**Packages, containers, runtimes**	**Verify** source • version • license • hash/signature • maintenance • approved use
Connectors	**Tools, MCP servers, plug-ins**	**Verify** source • version • license • hash/signature • maintenance • approved use

A model file from an unknown repository is not safer than an unknown binary.

Agent Tools: Permission Before Power



R Read-only by default

Claim status and policy lookup should not share write permissions.

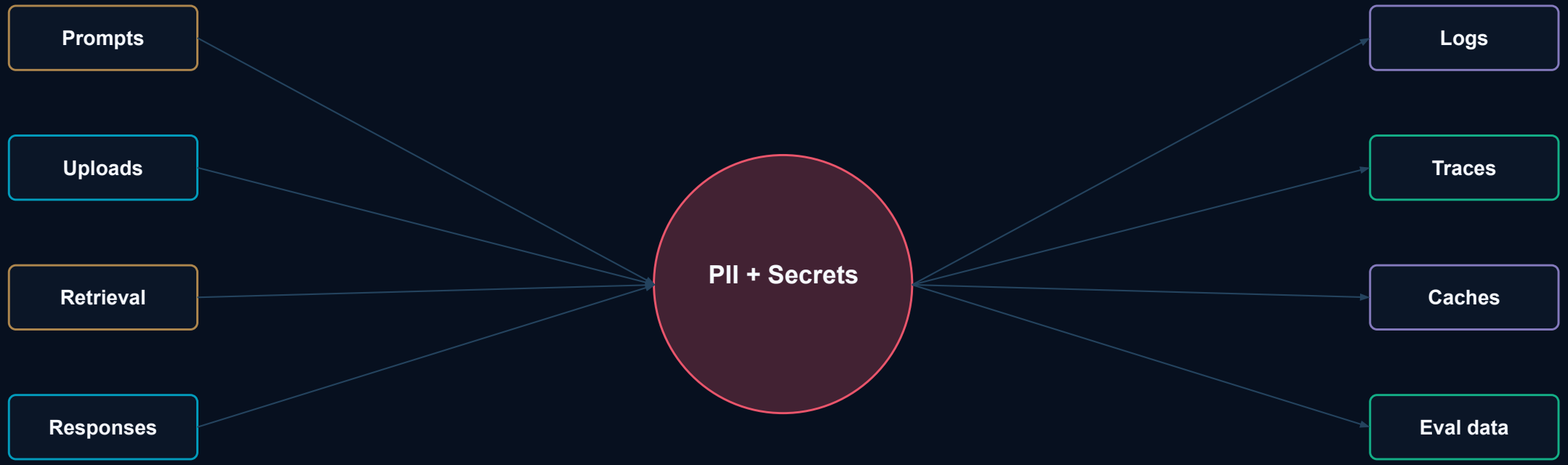
W Write actions need review

Follow-up tasks or messages should be approved by authorized staff.

! No arbitrary execution

The model should not generate SQL, shell, or infrastructure commands.

Sensitive Data: Control Every Leak Path



A prompt saying “do not reveal PII” is not a privacy program.

Improper Output Handling

Never trust generated output directly into:

- SQL statements
- Shell commands
- HTML / JavaScript
- File paths
- Admin APIs

Safer design pattern:

- Predefined actions
- Validated parameters
- Server-side checks
- Output encoding
- Human review for high impact

STOP RUNAWAY BEHAVIOR

Consumption & Availability Controls

1 **Request size**

Input limits

2 **Token budget**

Cost guard

3 **Step count**

Agent loop stop

4 **Tool calls**

Action limits

5 **Timeouts**

Safe failure

6 **Concurrency**

Capacity control

The safest agent is one that knows when to stop.

EVIDENCE

Testing Matrix: Security vs. Quality

Security tests

- Prompt injection
- Unauthorized retrieval
- Tool misuse
- Sensitive-data extraction
- Resource exhaustion

Model + workflow evaluation

- Correct retrieval
- Citations support answer
- Escalation behavior
- Refusal quality
- Human-review routing

A system can pass a quality benchmark and still be insecure.

AUTHORIZED TESTING

Red Teaming With Evidence

T Tools support—not certify

PyRIT, Garak, and Promptfoo can automate adversarial tests, but findings need human validation.

! Do not test without authorization

Define the target, environment, allowed actions, and evidence rules before running attacks.

E Report like an engineer

Include reproduction steps, impact, remediation owner, and retest evidence.

Red teaming is a controlled engineering process, not random jailbreak attempts.

Guardrails Are One Layer

Test guardrails for misses, false positives, latency, bypasses, and failure behavior.

CURRENT REFERENCES

Frameworks That Organize the Work

O OWASP LLM Top 10 2025

Prompt injection, sensitive disclosure, supply chain, poisoning, output handling, excessive agency, vector weaknesses, misinformation, unbounded consumption.

N NIST AI RMF + GenAI Profile

Govern, Map, Measure, and Manage AI risk; use NIST AI 600-1 for generative AI-specific risk considerations.

M MITRE ATLAS

Threat-model adversarial tactics, techniques, mitigations, and case studies for AI-enabled systems.

A OWASP ASVS / API / Cloud standards

AI security still depends on conventional application, API, cloud, identity, and software-supply-chain controls.

Record the version and review date because AI security guidance changes quickly.

Incident Response for AI Failures

Disable tool/model route

Containment and evidence

Revoke credentials

Containment and evidence

Preserve prompts + traces

Containment and evidence

Identify affected data

Recovery and verification

Remove poison / rebuild index

Recovery and verification

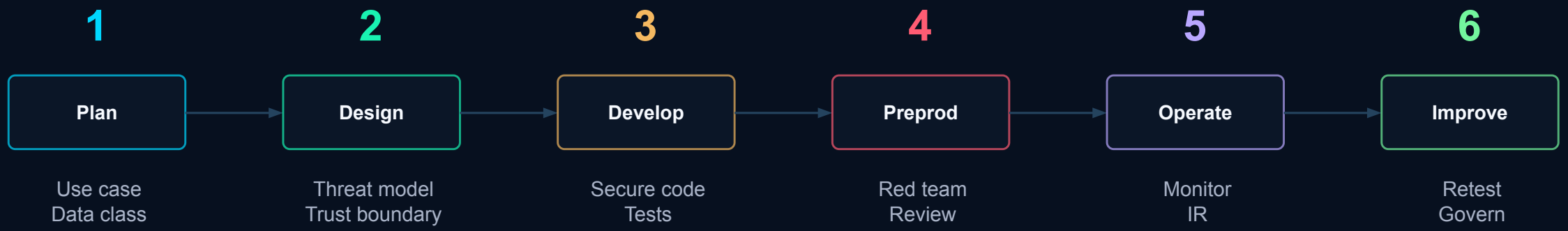
Retest and document

Recovery and verification

AI incidents must connect model behavior, application logs, tool actions, and business impact.

SHIFT-LEFT, STAY-RIGHT

Build Security Into the Lifecycle



- Security requirements before model selection
- Architecture review before production
- Monitoring and retesting after release

MARKET SIGNAL

U.S. Job-Market Reality Check

1,000+

broader AI cybersecurity results
LinkedIn U.S. snapshot • Aug 2, 2026

🔎 Search related titles

AI Security Engineer • GenAI Security Engineer • AI Red Team Engineer • AI/ML Security Engineer • Agent Security Engineer

✓ Reality check

Many production roles expect AppSec, cloud security, DevSecOps, security engineering, or ML platform experience.

! No guarantee language

Job-board counts are directional snapshots, not official vacancy totals or placement promises.

WORKBOOK + LABS

How ITLearn360 Helps You Build Evidence

Westbridge case study

A realistic claims-review AI system to secure

Hands-on templates

Data flow, threat model, permission matrix

Security labs

Secure RAG, prompt injection, tool misuse

Interview practice

Scenario Q&A and follow-up discussions

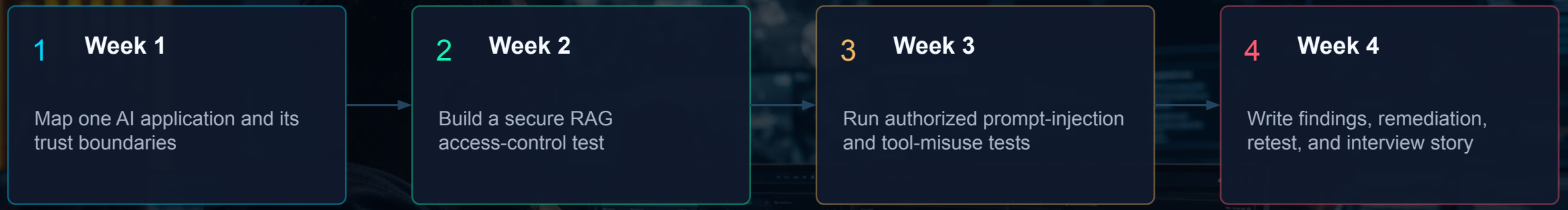
Portfolio artifacts

Findings report, retest evidence, IR playbook

The goal is not to memorize tools. The goal is to demonstrate how you identify, test, mitigate, and explain AI security risk.

CLOSING

Practical Action Plan



Secure the system — not just the prompt.

ITLearn360