

EXECUTIVE BRIEFING

AI Control Architecture

An open-source, vendor-agnostic model for governing, securing, assuring, and containing enterprise AI.

Make AI's authority bounded, provable, and reversible,
before you grant it, not after it fails.

AI is being adopted faster than control can absorb it.

1

Copilots are on

enabled across the workforce, often by default

2

AI is inside your SaaS

vendor features process enterprise data silently

3

Knowledge is wired in

RAG systems read corporate repositories

4

Agents are starting to act

calling tools, triggering workflows, updating records

Most of this happens before anyone can answer the basic control questions. An AI policy states intent; it cannot, by itself, bound what a specific system can see, decide, or do.

THE CORE THESIS

The next major AI failure in the enterprise will not come from a model becoming evil.

It will come from giving a **probabilistic system deterministic authority**, over data, decisions, or actions, **without a control architecture**.

Ankush Chowdhary Founder, Neo Control

Neither is dangerous alone. A probabilistic system generating inert text is low-risk. A deterministic system with real authority is normal software. It is the pairing, real authority and non-deterministic behavior, that existing controls were not designed for.

Six questions a policy can't answer

1

What AI exists?

inventory · ownership · classification

2

What can it see?

data boundaries · retrieval scope · access

3

What can it decide?

output validation · decision influence

4

What can it do?

tools · actions · workflows · agents

5

Who is accountable?

a named human owns every outcome

6

How is failure evidenced & contained?

monitoring · logging · containment · recovery

An AI touches the enterprise in exactly three ways

SEE

What data, context, and systems it can access.

Uncontrolled → confidentiality & boundary failure.

DECIDE

What decisions its output influences.

Uncontrolled → output silently becomes decision.

DO

What tools, actions, and workflows it can run.

Uncontrolled → the AI becomes an unbounded actor.

Risk rises as a use case moves See → Decide → Do, and as human oversight falls away. A read-only copilot and an action-capable agent are not the same control problem.

WHAT “UNDER CONTROL” ACTUALLY MEANS

A claim you can make for any AI you run

- We know what it can see, decide, and do.
- We have graded how strongly each boundary is held.
- A named human owns every consequential outcome.
- We can observe and reconstruct what it did.
- We can stop it, and we can recover.

BOUNDARY SOURCE: how you know a control is real

Declared

self-attested in the assessment

Evidenced

backed by config, docs, or registry

Verified

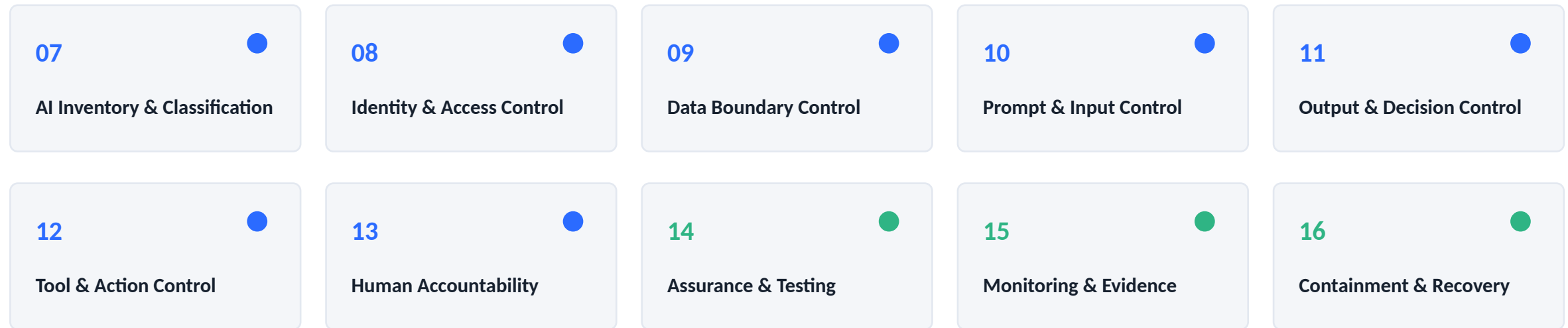
confirmed by a live check

Enforced

an active control point blocks it inline

Never lets “we have a policy” masquerade as “the control holds.”

Bound what AI is and can do, then prove it



- Pillars 07–13 bound what an AI is and can do
- Pillars 14–16 make that boundedness provable, observable, reversible

Controls are graduated by risk tier: a low-impact copilot and an action-capable agent do not carry the same burden.

Five tiers, not one process for all AI

T1	Fast-track	Low-risk productivity, public data only	Minimal
T2	Standard	Enterprise data or vendor AI	Data & vendor review
T3	Enhanced	Influences decisions & records	Decision accountability
T4	Action-capable	Calls tools, triggers workflows	Containment & approval gates
T5	High-impact	Autonomous, regulated, hard to reverse	Full assurance & evidence

Avoid both failures: over-controlling low-risk AI creates bureaucracy and avoidance; under-controlling high-risk AI creates unmanaged enterprise risk.

Every failure is a missing control, not a malicious model

An agent that cannot be stopped

no kill-switch, no revocation path

A RAG system surfaces data the user was never entitled to

boundary never enforced

An AI recommendation becomes a decision with no owner

accountability gap

A vendor AI retains prompts you assumed were transient

unreviewed processing

An incident that cannot be reconstructed

logging insufficient

An exception that quietly became permanent

no expiry, no review

The common control language beneath the frameworks

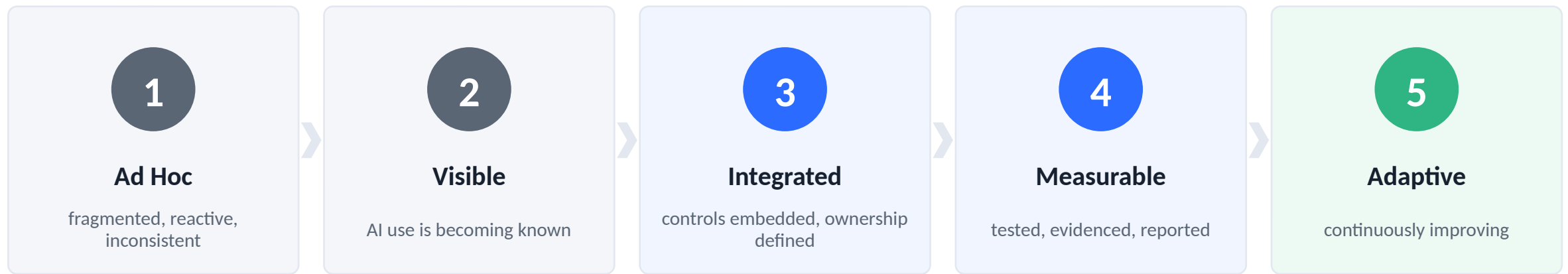
ACA is not a rival to the recognized frameworks. Every control carries a conceptual crosswalk to the standards you are already held to, so one assessment can serve many obligations.

NIST AI RMF Govern · Map · Measure · Manage	ISO/IEC 42001 AI management system	EU AI Act classification · risk · logging · transparency
SR 11-7 model risk management	NYDFS Part 500 cybersecurity program & access	OWASP LLM & Agentic injection · disclosure · excessive agency

Crosswalks are conceptual, not legal opinions.

WHERE ARE YOU TODAY?

A maturity path, not a big-bang program



The goal at every level is the next one: move from unknown AI use to known, then to controlled, then to proven. Start where you are.

Start with one use case, end to end

1

Pick one real AI use case

a copilot, a RAG assistant, or an agent already in use

2

Classify & risk-tier it

See / Decide / Do → Tier 1-5

3

Select the controls for that tier

graduated, not the whole catalogue

4

Collect evidence & grade the boundary

Declared → Verified → Enforced

5

Prove it, then repeat

one assessment, reusable across obligations

OPEN, SO YOU CAN START FREE

- The specification is open (CC BY 4.0)
- Reference assessment code is open source
- A quickstart walks one use case end to end
- No vendor required to adopt it

aicontrolarchitecture.org

OPEN SOURCE · VENDOR-AGNOSTIC · STEWARDED BY NEO

The architecture is open. Neo is the strongest way to operate it.

ACA stays vendor-neutral and free to adopt. Neo Control stewards the specification and offers the reference implementation, turning the same controls into live assessment, verification, evidence, and enforcement at scale.

***Make AI's authority bounded, provable, and reversible,
before you grant it, not after it fails.***