

ELECTRIC POWER
STEERING &
STEER-BY-WIRE

COLUMNS &
INTERMEDIATE SHAFTS

DRIVELINE SYSTEMS

BRAKE-BY-WIRE

SOFTWARE

HYDRAULIC POWER
STEERING



Intelligent AI in Safety-Critical Embedded Software Development

Copyright 2026, Nexteer Automotive Corporation. All rights reserved.

Nexteer Confidential

Authors and Contributors

Intelligent AI in Safety-Critical Embedded Software Development

PRESENTER



Bobby O'Steen

Senior Software
Engineering Manager

Nexteer Automotive

Aachen presenter

CO-AUTHORS

- **Owen Tosh**
- **Noah Reinke**
- **Sunny Judge**
- **Michael Story**

Nexteer Automotive

TECHNICAL CONTRIBUTION & REVIEW



Vinod Shankar

AI-assisted software-
development methods,
engineering controls,
workflow considerations,
and technical review

Nexteer Automotive

Developed through collaboration across embedded software engineering, product development, and AI-assisted engineering activities.

Nexteer Automotive

Accelerating Mobility to be Safe, Green and Exciting



World headquarters in
Auburn Hills, Michigan, USA, with
more than **40 locations worldwide**

Global Workforce:
~12,600



MOTION CONTROL SOLUTIONS

- **ELECTRIC POWER STEERING & STEER-BY-WIRE**
- **COLUMNS & I-SHAFTS**
- **DRIVELINE**
- **BRAKE-BY-WIRE**
- **SOFTWARE**
- **HYDRAULIC POWER STEERING**

The Responsibility Behind the Answer

A release decision is simple to state, but complex to support.

AT THE RELEASE REVIEW

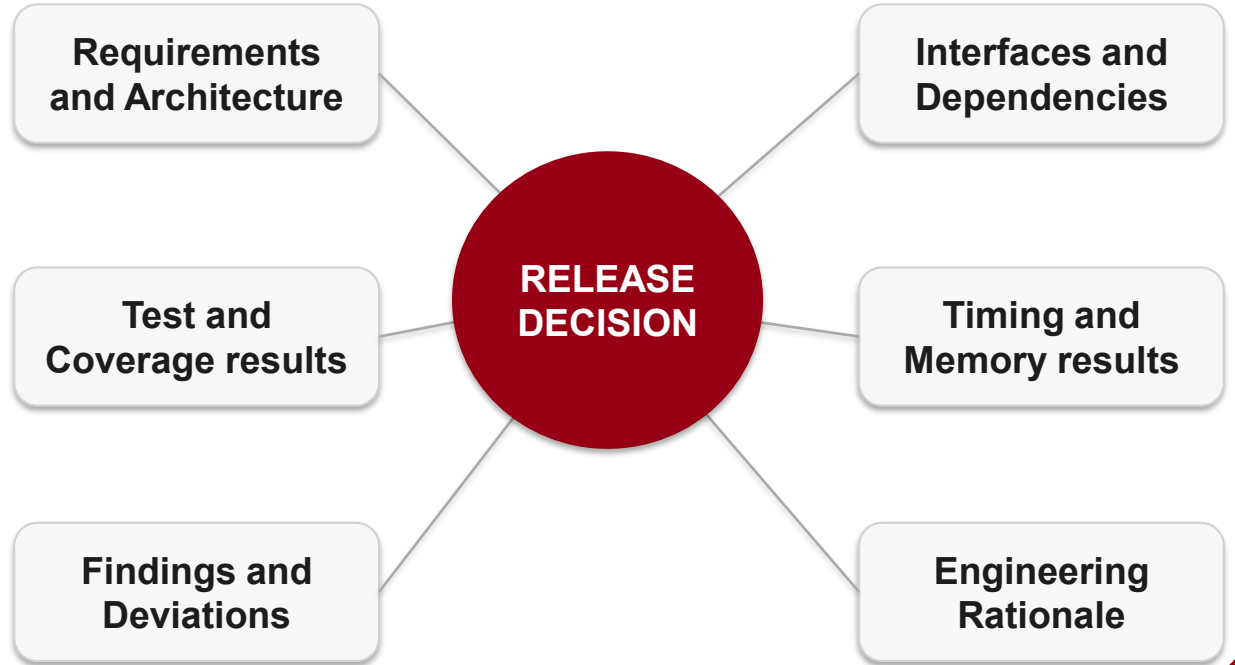
“Is embedded software ready?”

**As the embedded-software representative,
I must answer with confidence.**

The answer carries engineering judgment, risk disposition, and accountability.



EVIDENCE BEHIND THE ANSWER



The decision remains human. The evidence behind it continues to grow.

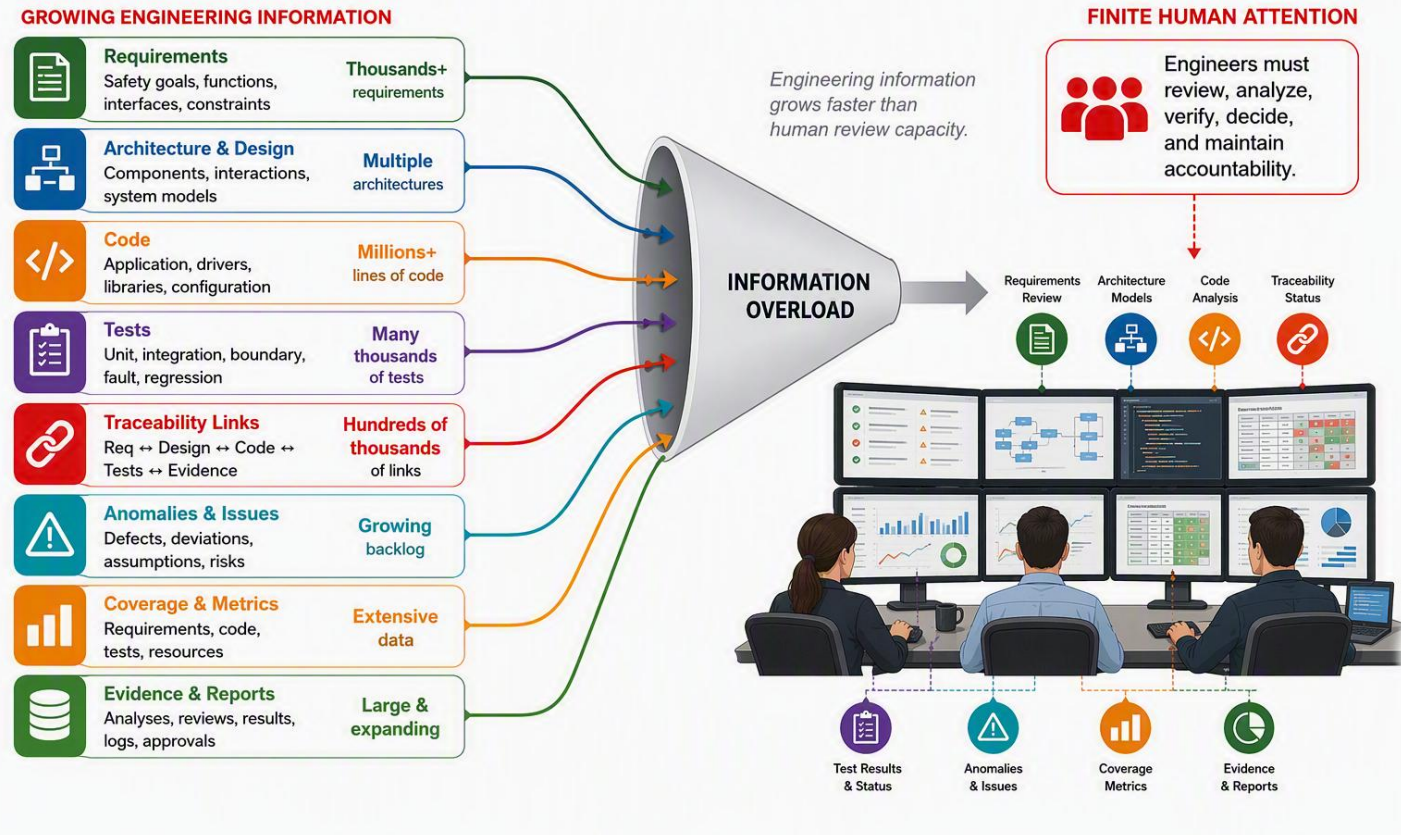
How can AI extend engineering capacity without weakening the rigor required for safety-critical software?

Why AI Matters Now

AI is attractive because software, evidence, and release decisions are growing faster than manual review capacity.

The Problem: Engineering Scale Exceeds Human Attention

AI is attractive because software, evidence, and release decisions are growing faster than manual review capacity.



ENGINEERING PRESSURE

- **More complexity**
Functions, interfaces, dependencies, and variants
- **Faster change**
Shorter delivery cycles and more frequent updates
- **More evidence**
Tests, traceability, analyses, anomalies, rationale
- **Finite attention**
Critical knowledge is not always available at decision time

AI may extend engineering capacity. It does not replace engineering judgment or release accountability.

Which AI Are We Discussing?

Different AI applications create different engineering risks and evidence needs.

PREDICTIVE AI

Learns relationships from historical and labeled data

- Classification and detection
- Prediction and forecasting
- Risk and anomaly identification
- Typical focus: data, performance, trends, and generalization

GENERATIVE AI

Creates proposed engineering content

- Requirements, code, tests, analyses
- Hallucinations, context and provenance
- Human review before acceptance

AGENTIC AI

Plans and performs steps using approved tools

- Retrieves context and invokes tools
- Modifies artifacts or runs workflows
- Permissions, boundaries, auditability

FOCUS OF THIS TALK: AI used within the engineering development process, rather than AI embedded within vehicle functionality.

AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.



AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.

AI Creates & Derives

Requirements

Architecture

Design & Code

Tests

Analysis

**Output still
requires human
review**

AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.

AI Creates & Derives

Requirements

Architecture

Design & Code

Tests

Analysis

**Output still
requires human
review**



Potential Failures & Contributing Factors

Hallucinations

Inconsistent output

Missing context

Weak traceability

Prompt injection / data
exposure

Automation bias

AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.

AI Creates & Derives

Requirements

Architecture

Design & Code

Tests

Analysis

**Output still
requires human
review**



Potential Failures & Contributing Factors

Hallucinations

Inconsistent output

Missing context

Weak traceability

Prompt injection / data
exposure

Automation bias



Engineering Controls

Use-case classification

Approved context &
inputs

Review & analysis

Deterministic
verification

Traceability

Release governance

AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.

AI Creates & Derives

Requirements

Architecture

Design & Code

Tests

Analysis

**Output still
requires human
review**

Potential Failures & Contributing Factors

Hallucinations

Inconsistent output

Missing context

Weak traceability

Prompt injection / data
exposure

Automation bias

Engineering Controls

Use-case classification

Approved context &
inputs

Review & analysis

Deterministic
verification

Traceability

Release governance



Example: An AI-generated peripheral driver may compile successfully while using an incorrect hardware register value or omitting a required timeout.

AI Adds Capacity and New Failure Modes

Plausible output can still be incomplete, incorrect, insecure, inconsistent, or difficult to verify.

AI Creates & Derives

Requirements

Architecture

Design & Code

Tests

Analysis

**Output still
requires human
review**

Potential Failures & Contributing Factors

Hallucinations

Inconsistent output

Missing context

Weak traceability

Prompt injection / data
exposure

Automation bias

Engineering Controls

Use-case classification

Approved context &
inputs

Review & analysis

Deterministic
verification

Traceability

Release governance



Example: An AI-generated peripheral driver may compile successfully while using an incorrect hardware register value or omitting a required timeout.

AI output is a proposal. Engineers must review, verify, trace, and approve.

Learn Before Scaling

AI adoption should evaluate risk controls and evidence quality, not only whether AI produces output faster.



Engineering Measures: human correction rate • defects found • review time • regression results • token & compute cost

Controlled adoption, not uncontrolled experimentation.

Classify the Use, Then Set the Evidence

The safety argument begins with the intended use and error path, not with the model name.

1

INTENDED USE

What activity is permitted?

2

OUTPUT ARTIFACT

Requirement, code, tests, or analyses?

3

POTENTIAL IMPACT

Can it introduce or miss an error?

4

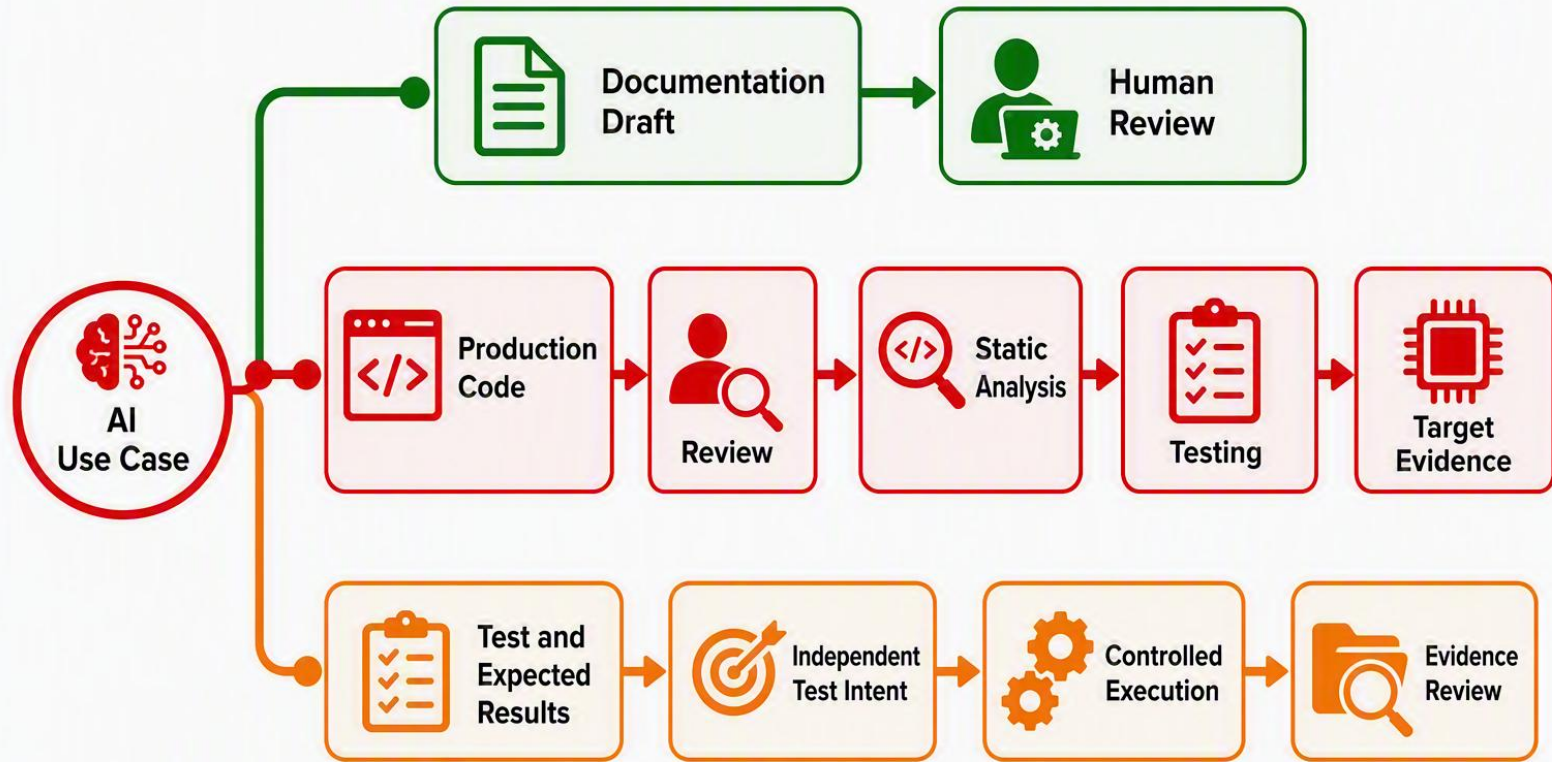
DETECTION CONTROLS

What independent mechanism detects it?

5

CONFIDENCE DECISION

What evidence or qualification is needed?



Classify the defined use case and error path before deciding what confidence evidence is needed.

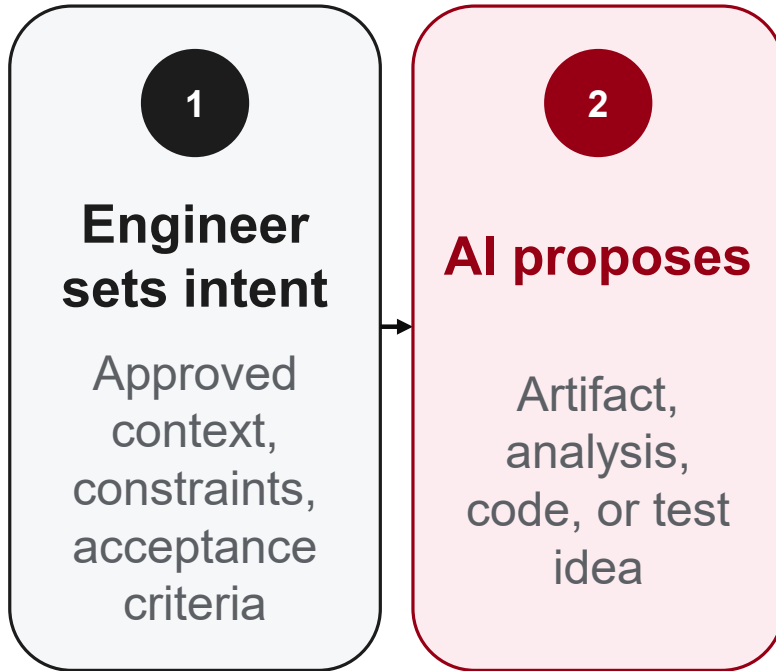
Humans Own the Decision

Human oversight is continuous, not a final proofreading checkpoint.



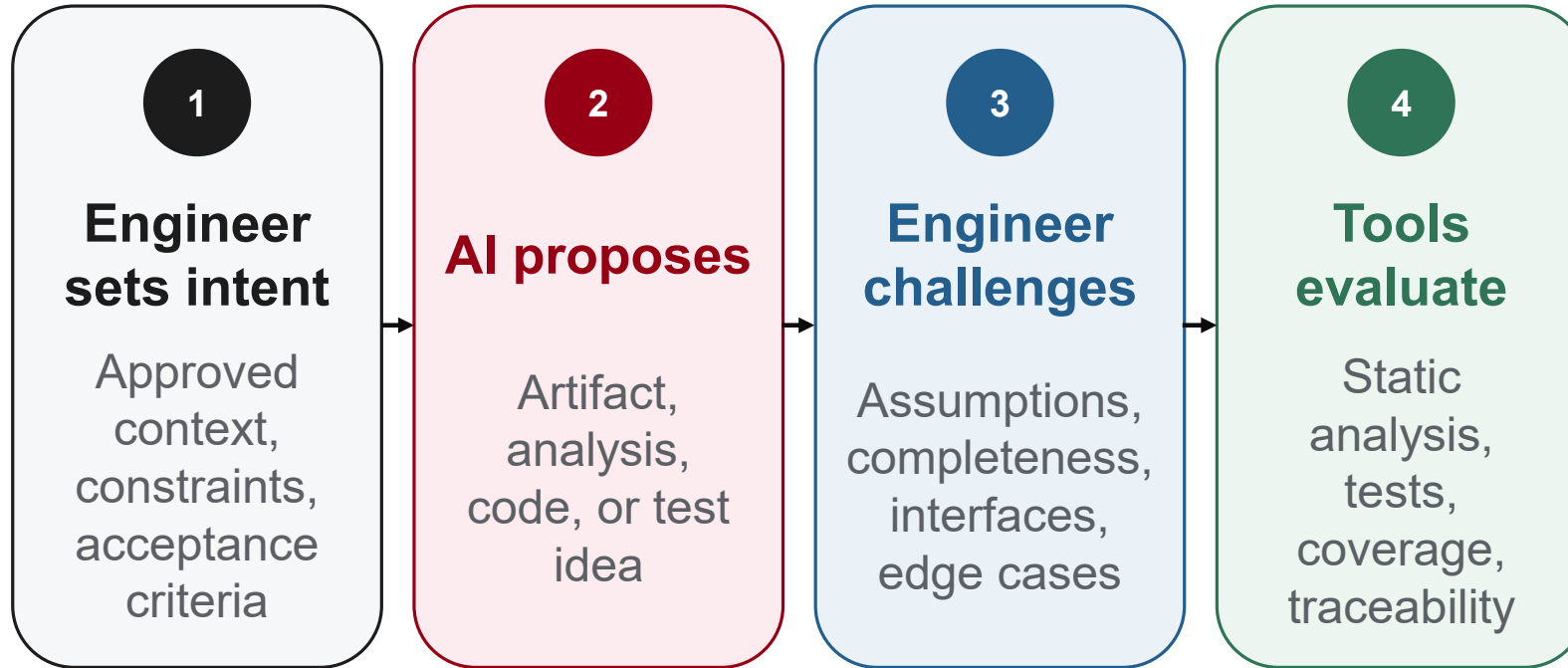
Humans Own the Decision

Human oversight is continuous, not a final proofreading checkpoint.



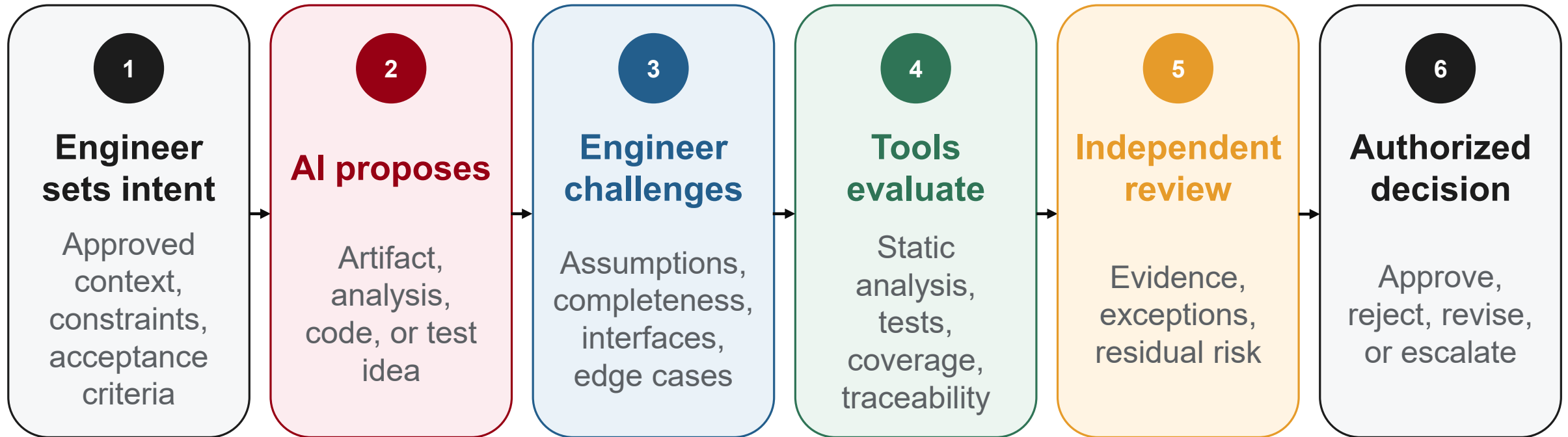
Humans Own the Decision

Human oversight is continuous, not a final proofreading checkpoint.



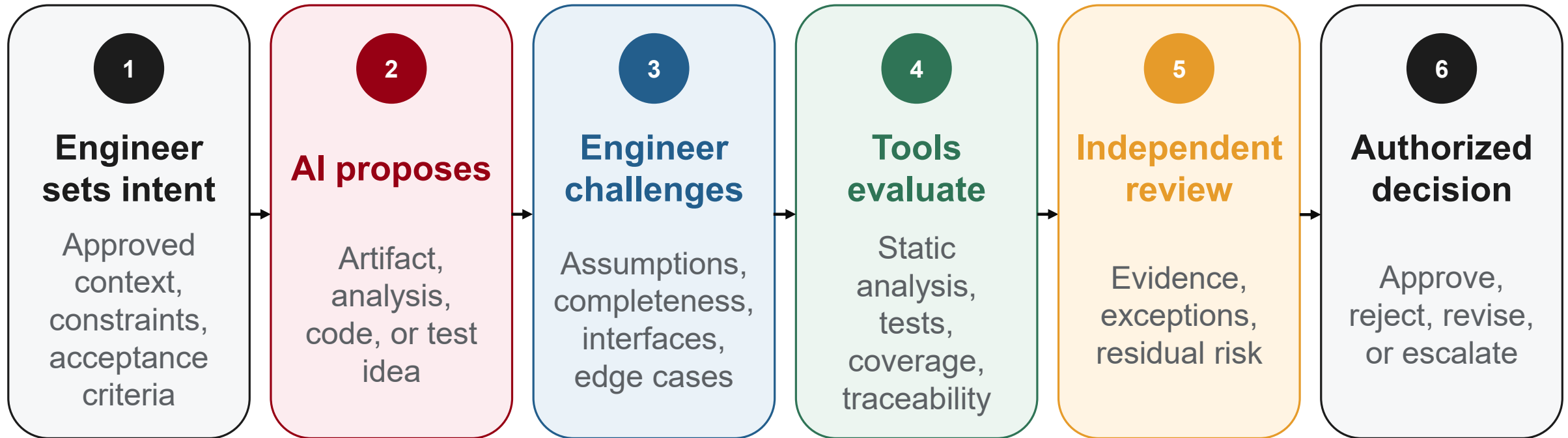
Humans Own the Decision

Human oversight is continuous, not a final proofreading checkpoint.



Humans Own the Decision

Human oversight is continuous, not a final proofreading checkpoint.



Example: An engineer reviewing an AI-generated CAN driver must validate register initialization, timing assumptions, error handling, and target behavior.

AI does not accept product risk, approve safety decisions, or authorize release.

Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.



Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.

IMPROVE INPUTS

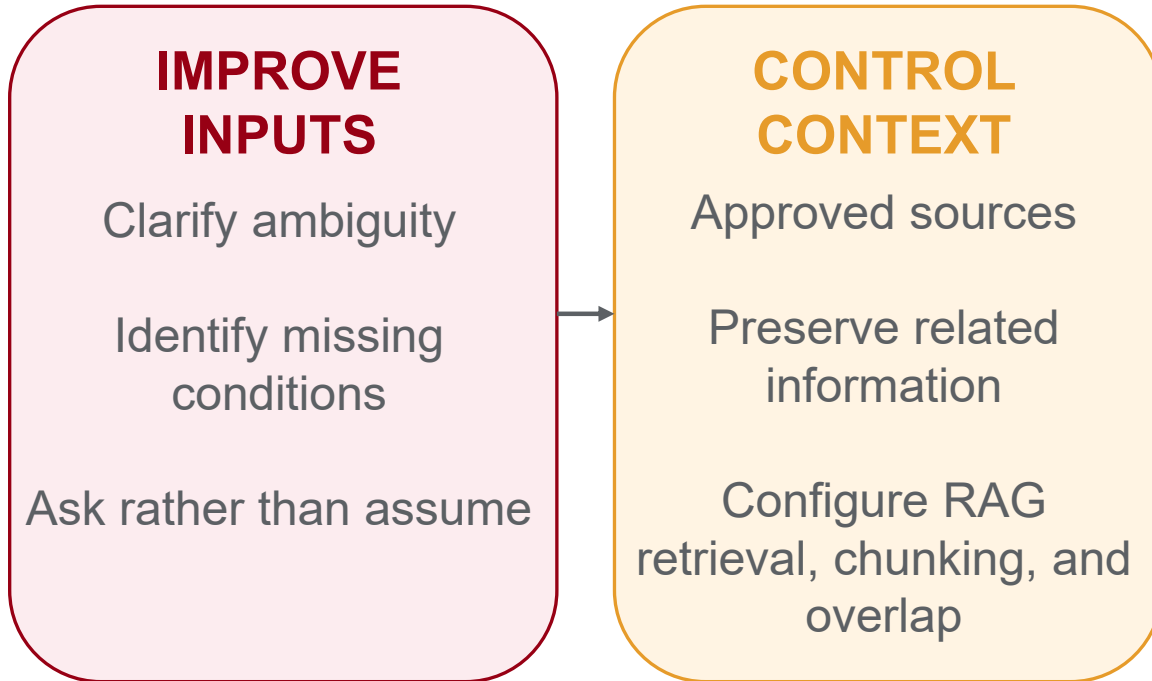
Clarify ambiguity

Identify missing
conditions

Ask rather than assume

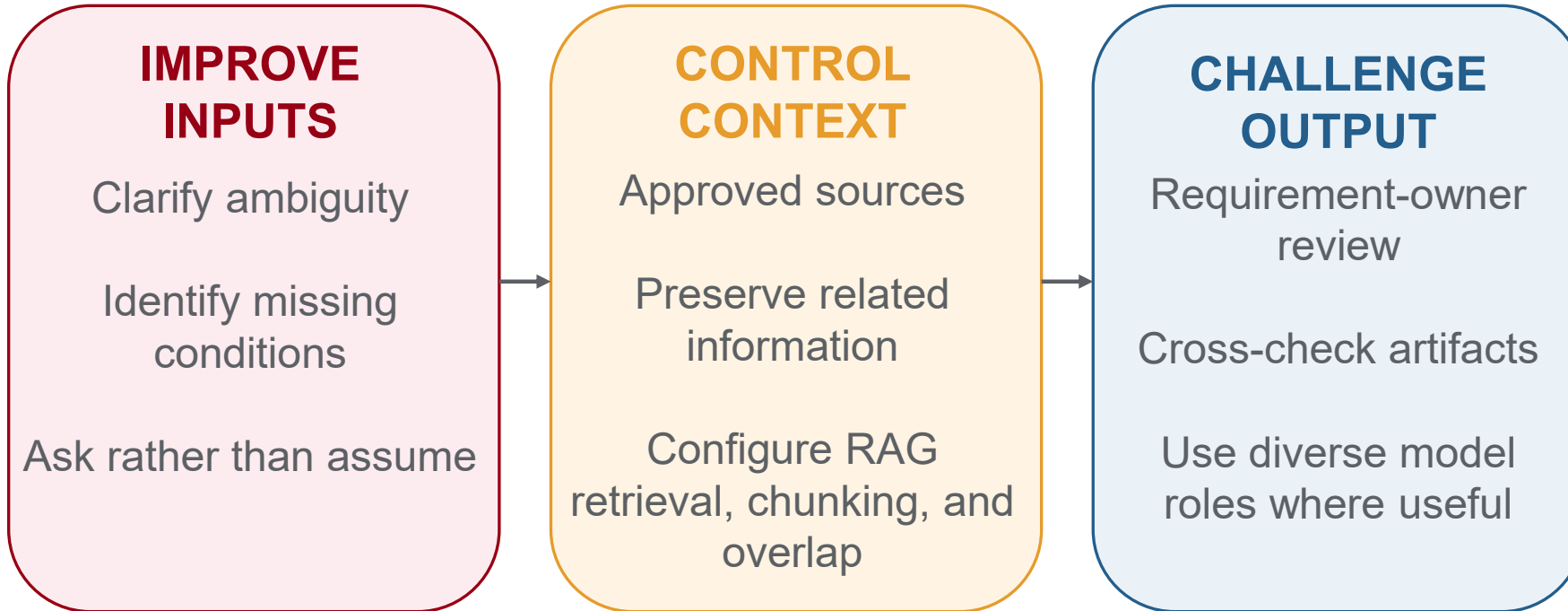
Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.



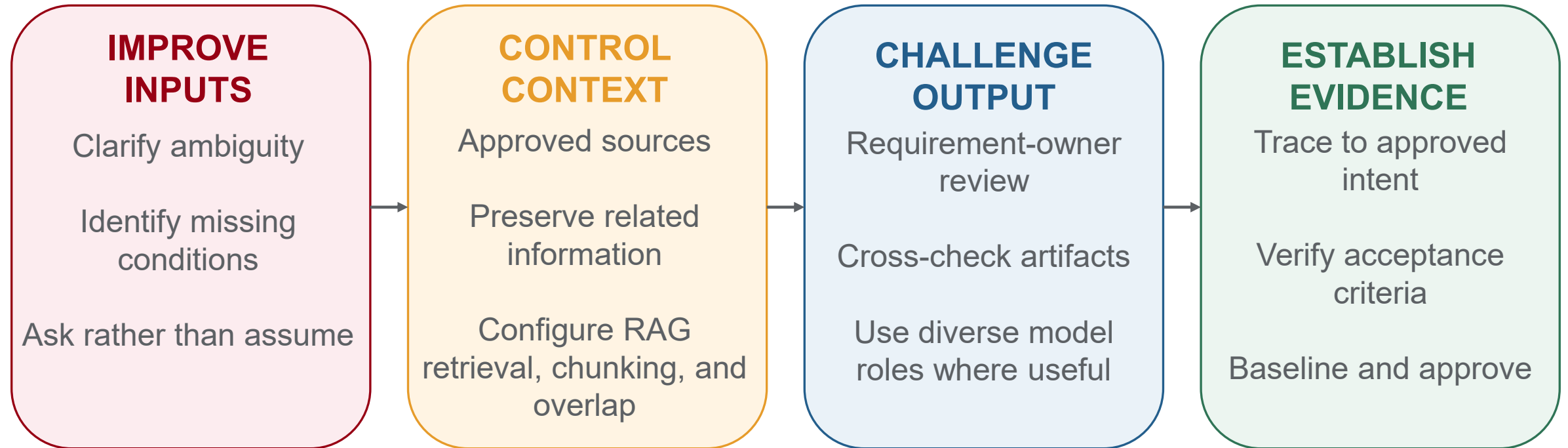
Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.



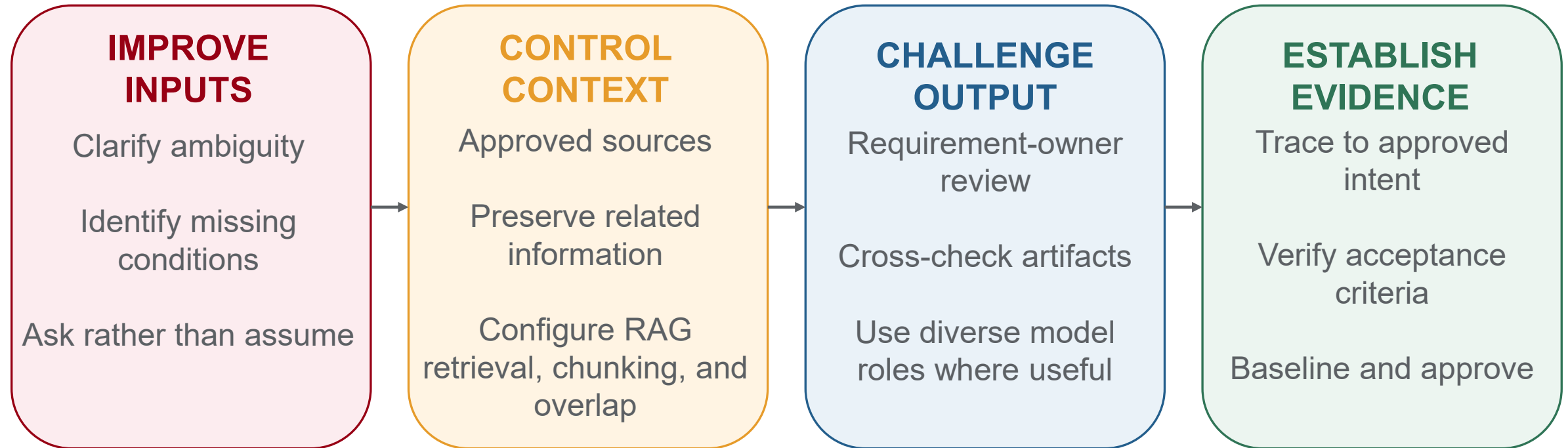
Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.



Reduce Hallucinations Before They Start

The most dangerous output may be plausible, internally consistent, and still wrong with respect to intent.



Example: If a torque-limiting requirement omits behavior for invalid sensor confidence, the **AI should ask for clarification** rather than invent a fallback.

Better output begins with approved intent, attributable context, and questions before assumptions.

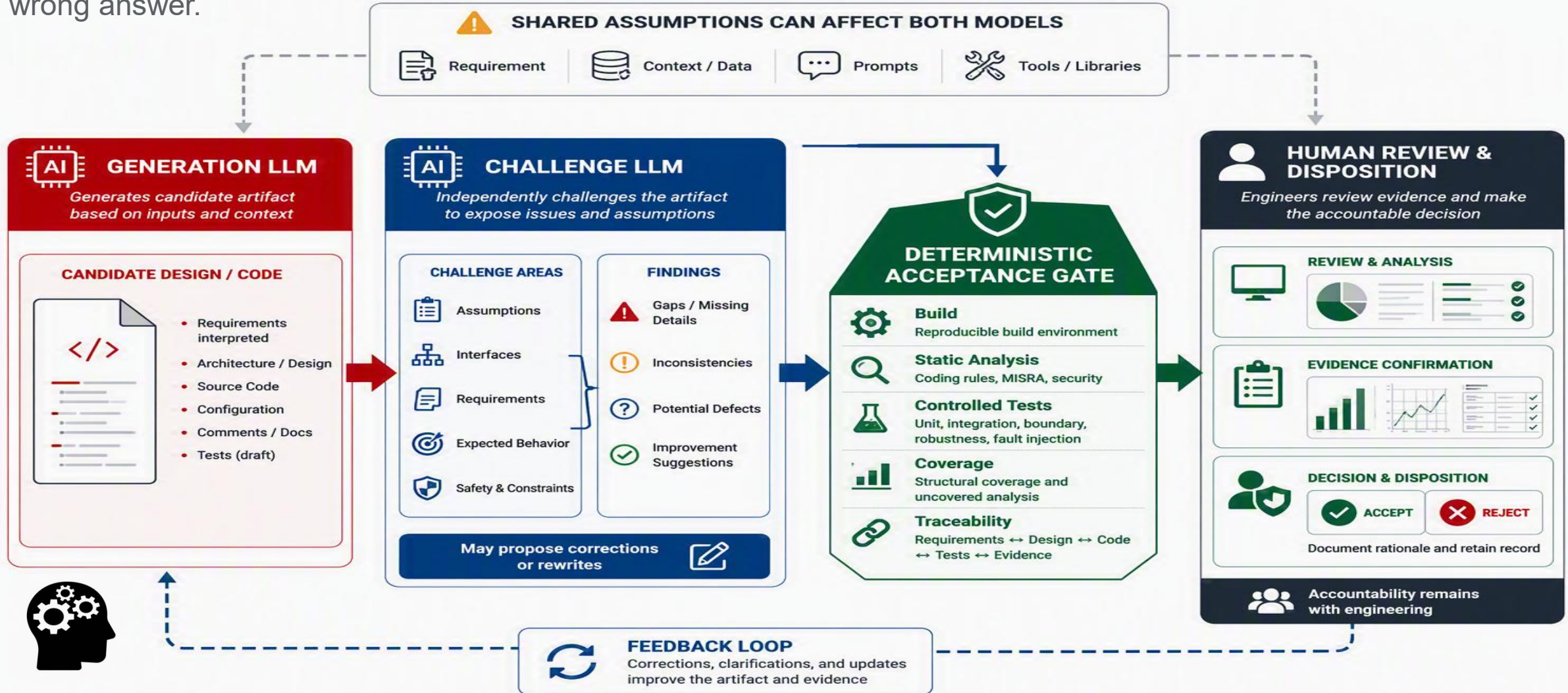
Independence Is More Than a Second LLM

Diverse models can challenge assumptions, but shared context and correlated training can still produce the same wrong answer.



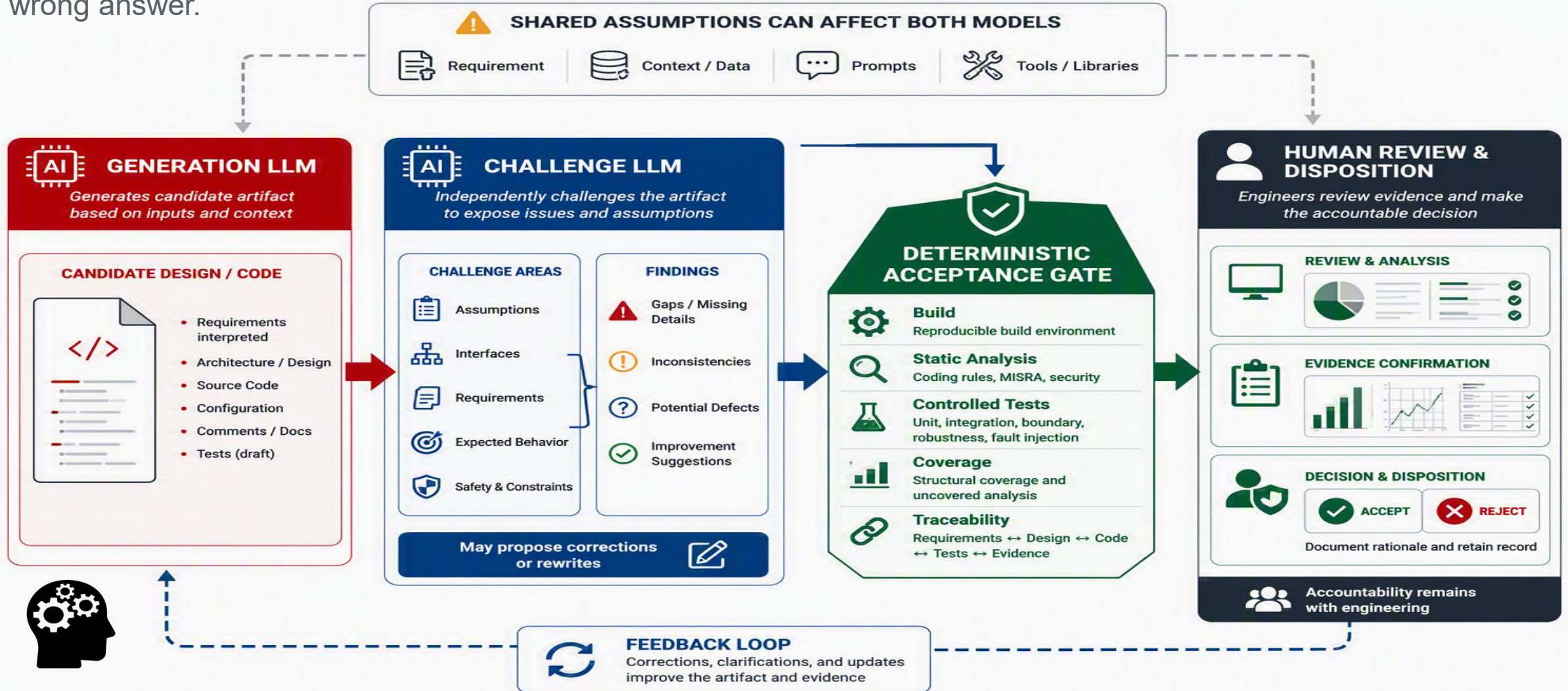
Independence Is More Than a Second LLM

Diverse models can challenge assumptions, but shared context and correlated training can still produce the same wrong answer.



Independence Is More Than a Second LLM

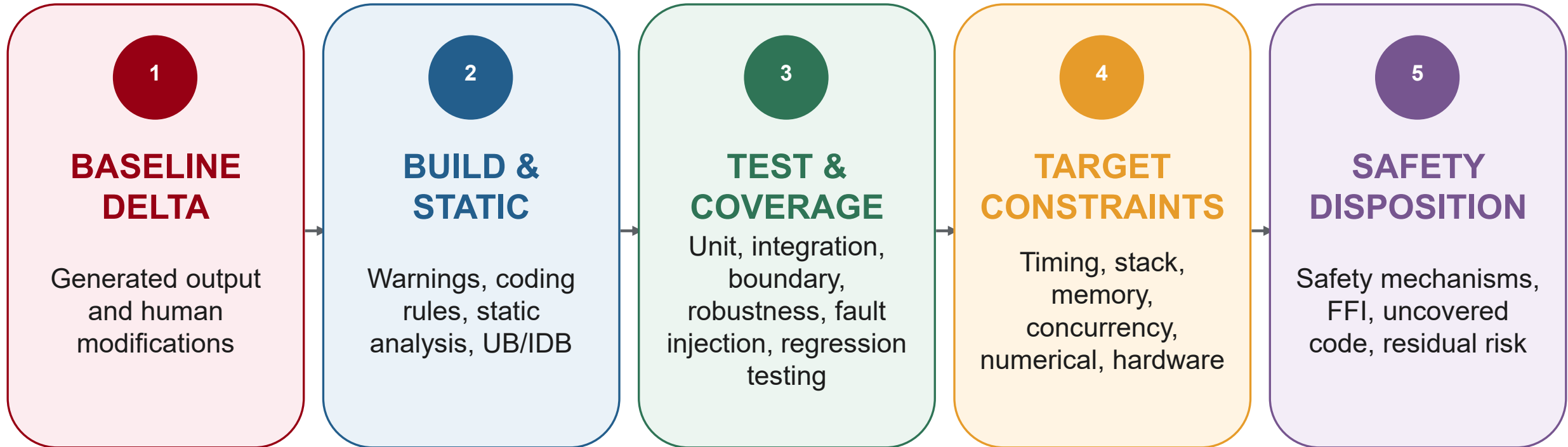
Diverse models can challenge assumptions, but shared context and correlated training can still produce the same wrong answer.



Model agreement supports review. Independent engineering evidence establishes confidence.

Freeze the Artifact, Then Verify It

Generation may be nondeterministic. Acceptance must be based on reproducible properties of the frozen artifact.

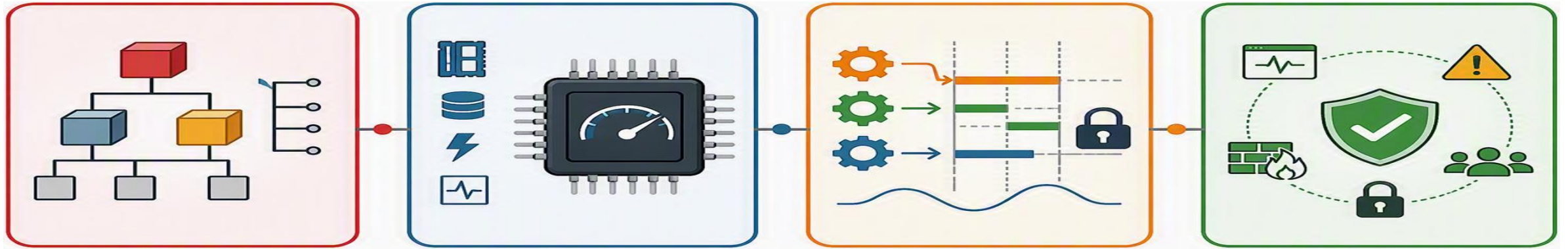


Reject or escalate when required evidence is missing, findings remain unresolved, or target constraints cannot be demonstrated.

A generated artifact is accepted on evidence, not plausibility or compilation alone.

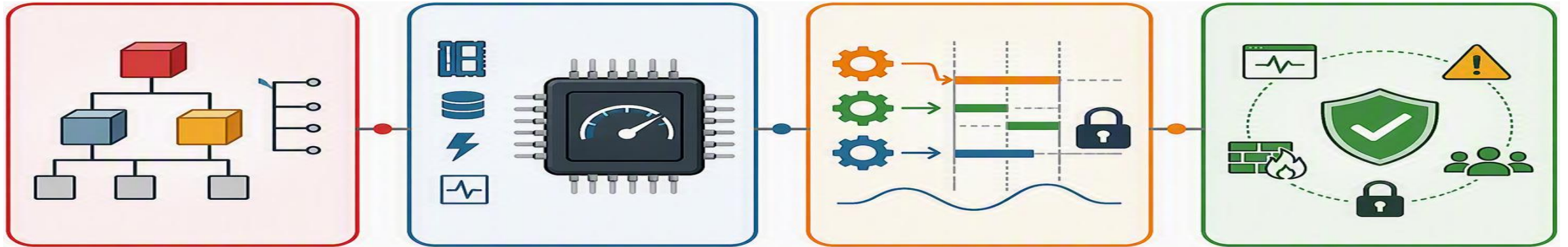
Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.



Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.

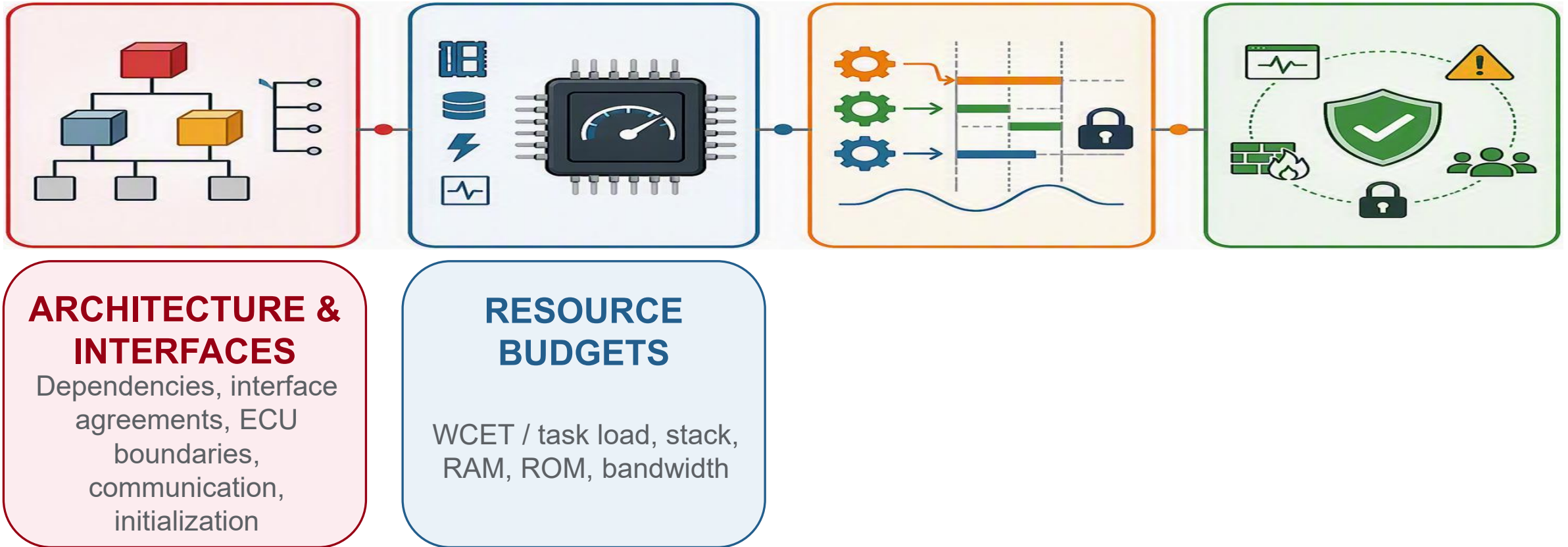


ARCHITECTURE & INTERFACES

Dependencies, interface agreements, ECU boundaries, communication, initialization

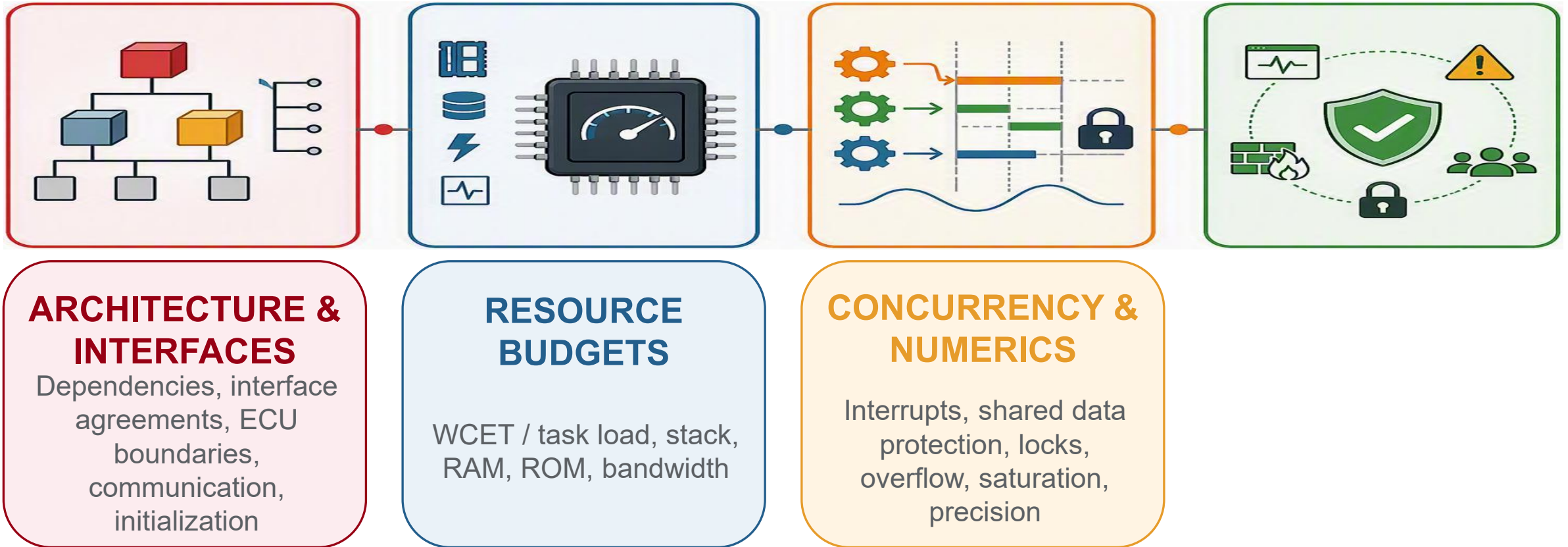
Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.



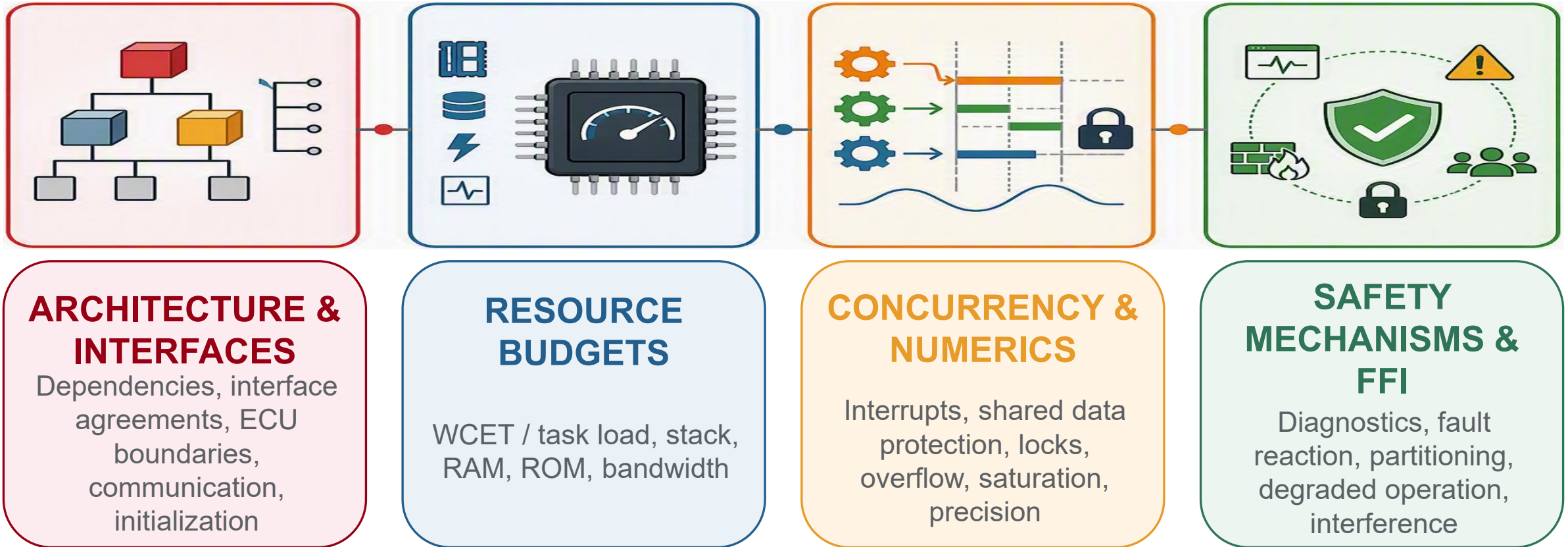
Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.



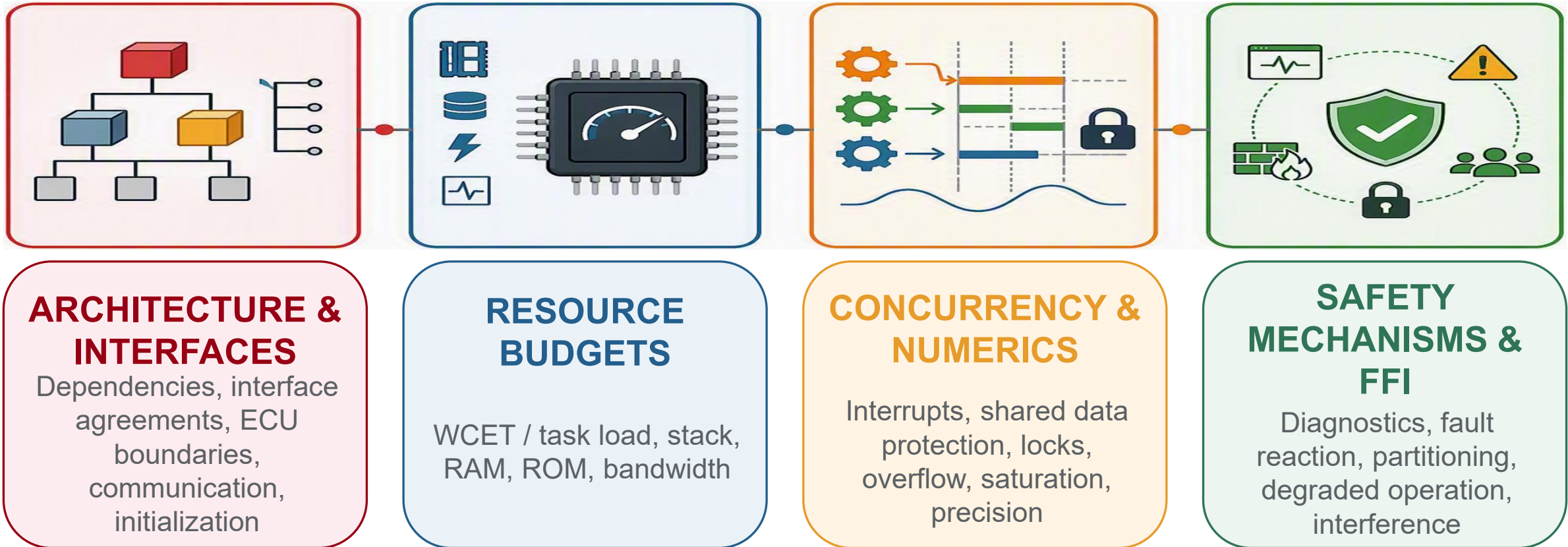
Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.



Verify the Real Embedded Target

Safety-related software is constrained by architecture, hardware, timing, numerical behavior, and fault responses.



Example: An AI "cleanup" of a motor-control function may preserve nominal behavior while changing execution time, memory usage, edge-case handling, or diagnostic responses.

Refactoring claims do not remove the need for architecture, resource-budget, safety-mechanism, and change-impact analysis.

Preserve the AI Evidence Record

A reviewer must be able to reconstruct what was generated, changed, verified, and approved.



CHANGE IMPACT

Requirements, architecture, safety analyses, resource budgets, tests

VERIFICATION RECORD

Build, analyses, tests, coverage, target results, anomalies, deviations

PROVENANCE & SECURITY

Approved sources, confidentiality, licensing, dependencies, permissions

APPROVAL RECORD

Competency, independence, residual risk, release authority, escalation



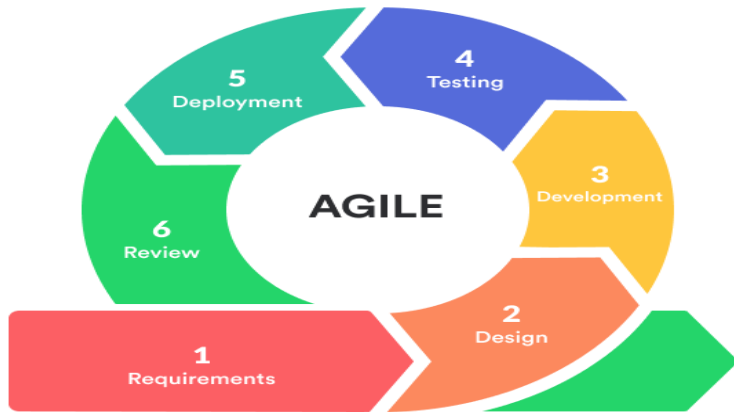
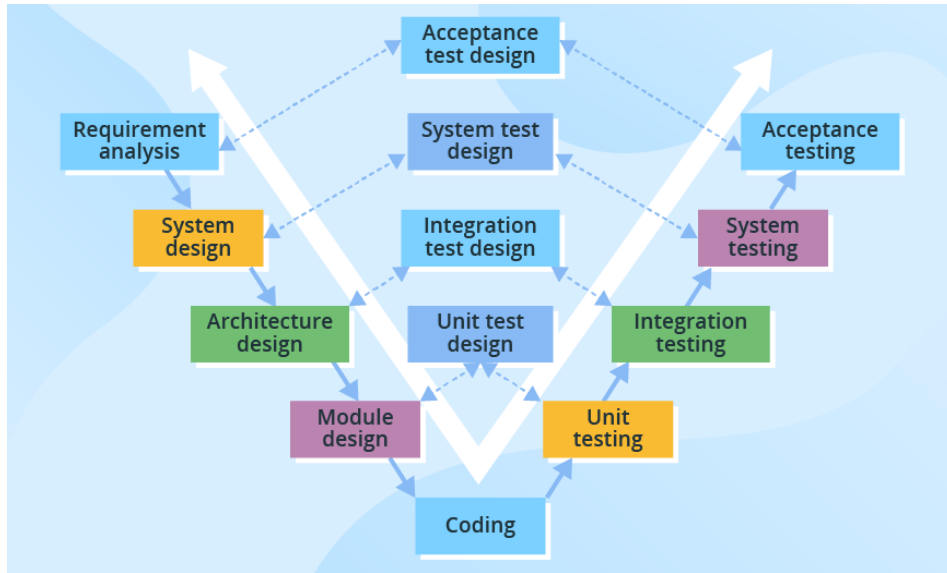
REASSESS when model, prompt, source context, permissions, tools, workflow, or accepted output changes.

If provenance, required verification, safety impact, or approval cannot be established: reject or escalate.

AI May Change Where Work Happens

Engineering workflows may evolve around AI capabilities while preserving verification, accountability, and release authority.

Human-Centric Development Models



HOW AI MAY CHANGE THE WORKFLOW

Work Across Traditional Boundaries

Requirements clarification, design, implementation, and test generation may overlap

Use AI Where It Performs Best

Do not force AI to reproduce workflows designed around human constraints

Preserve Engineering Control

Humans and controlled tools verify evidence, resolve risk, and authorize release

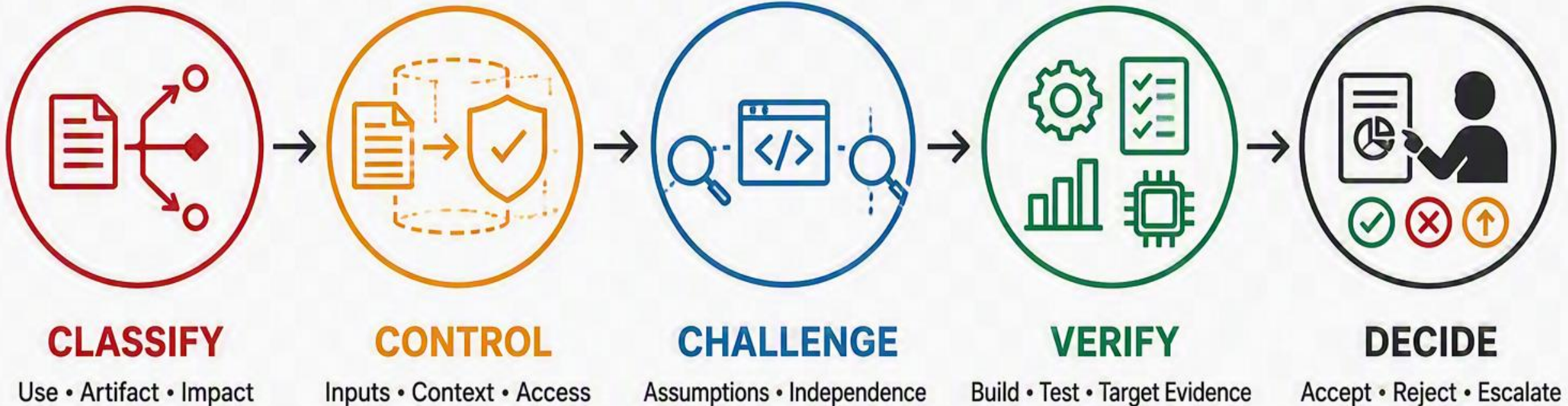


Example: AI can generate test ideas during design and implementation, crossing traditional lifecycle boundaries. Engineers execute and approve tests through controlled engineering processes.

Optimize for the engineering objective, not the traditional activity. Engineering accountability remains.

Five Controls for Responsible AI

Responsible AI use combines bounded application, controlled context, independent challenge, objective evidence, and human accountability.



Better engineering capability, without weakening the controls on which safe software depends.

Questions and Discussion

Practical adoption depends on intended use, governance, verification, and human ownership.

Where can AI create meaningful engineering value?

What evidence is needed before AI-assisted output can be accepted?

Where must human judgment and release authority remain explicit?

**The goal is not to remove engineers from the process.
The goal is to give engineers better tools while preserving the
judgment and accountability on which safe software depends.**



Learn More About Modelware
nexteer.com/software/

