
ELISA
Enabling Linux in
Safety Applications
ELISA Project Workshop
June 9-11, 2026
London, UK
Co-Hosted by: Canonical

NVIDIA

ELISA
Enabling Linux in Safety Applications

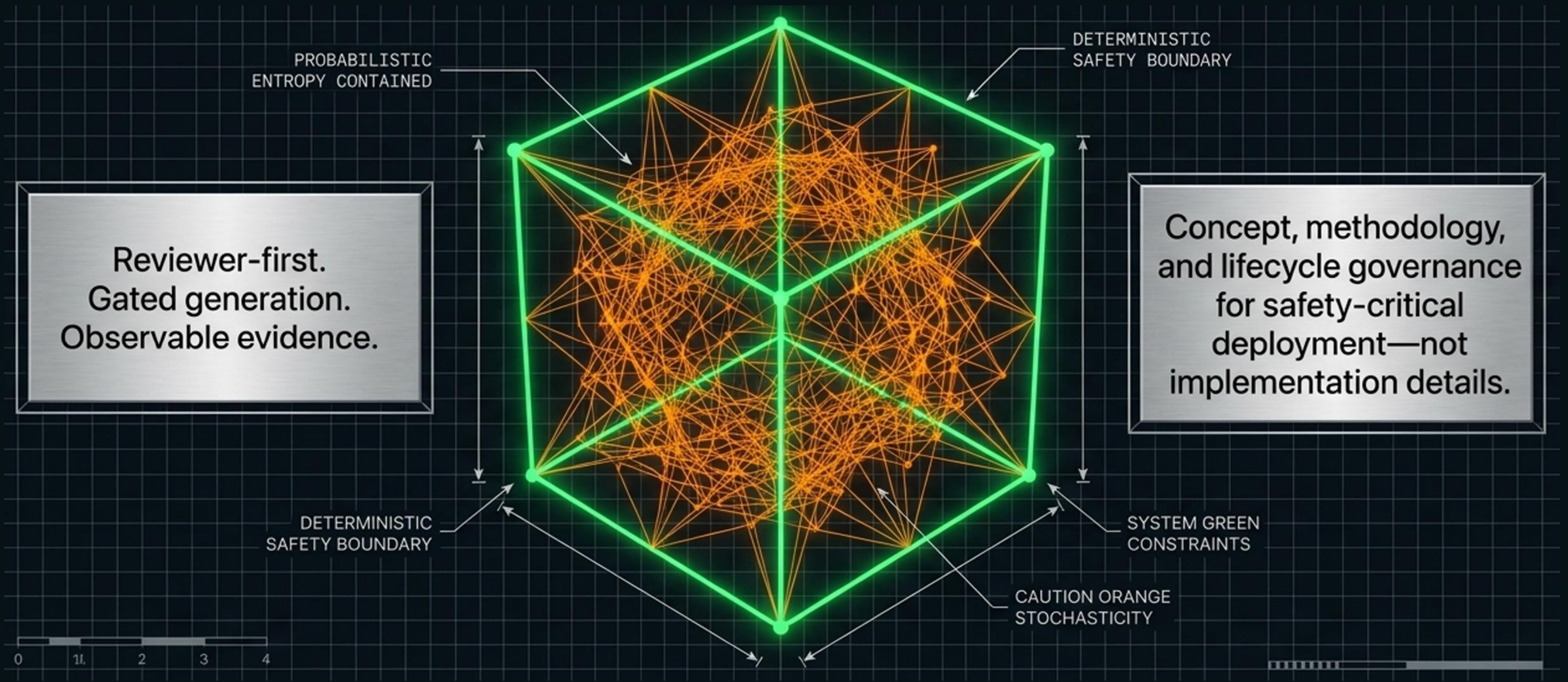
WORKSHOP

Applying AI to safety-critical product development

Path to Safe Generative AI Adoption in Regulated Product Lifecycles - Methodology and Examples

Nicola Di Miscio ndimiscio@nvidia.com, Francesco Saracino fsaracino@nvidia.com | v1.0 | 10 June 2026

Content



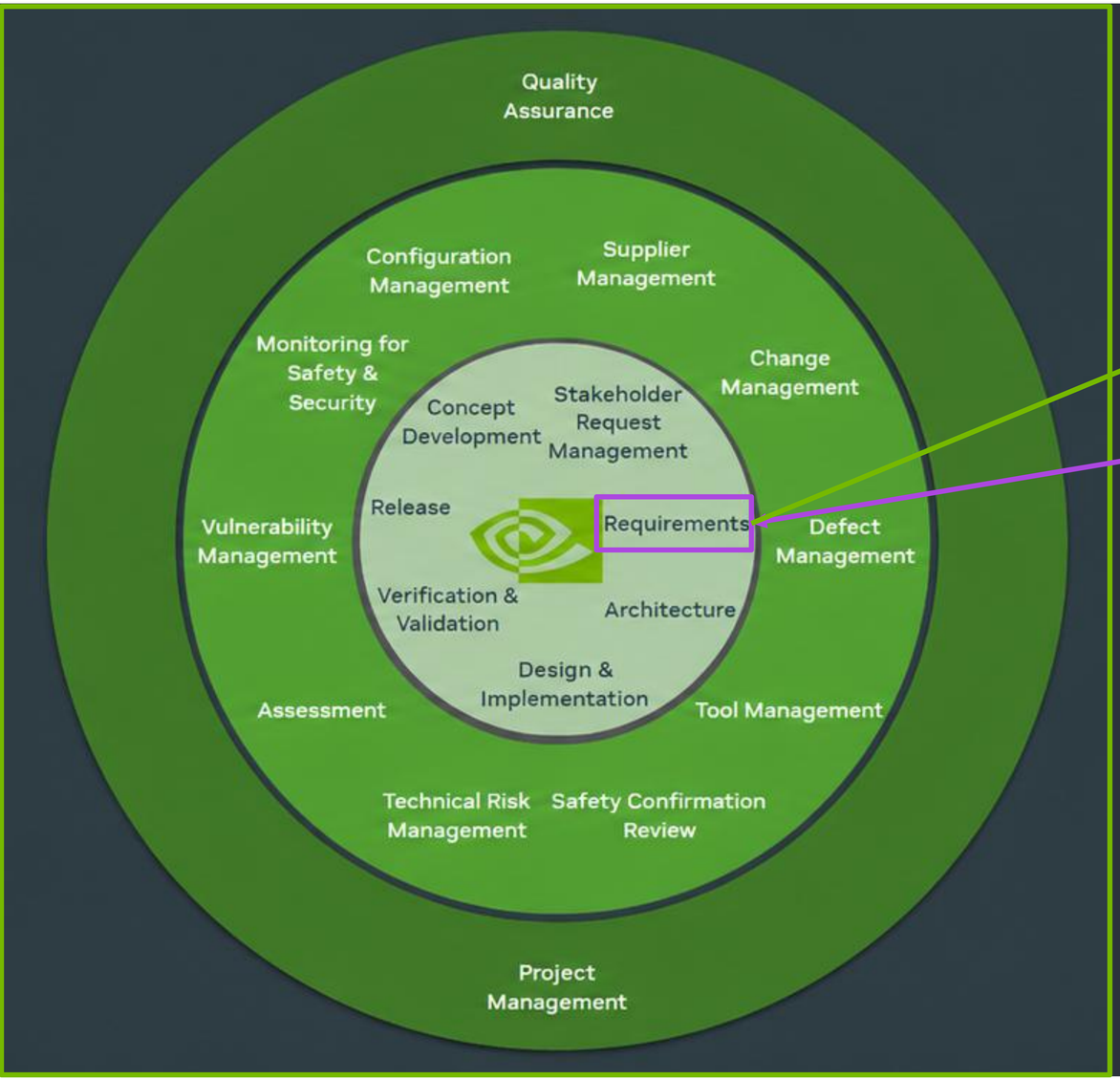
Our quest can be described in many ways all equally significant:

- **Human-over-AI-over-AI Governance Stack**
- **AI-Gated Generation System**
- **Reviewer-Gated AI Pipeline**
- **Dual-Layer AI Governance System** (inner generators, outer reviewers)
- **AI-Reviewed Generation Framework**

ELISA Enabling Linux in Safety Applications WORKSHOP

Deterministic AI Reviewers for DriveOS SRS

From GenAI safe deployment principles to an ISO 26262-classified Reviewer tool



The model may be non-deterministic; the governance had better not be.

Deterministic Input and Process

Non-Deterministic AI Output

AI Augmented Review of SRS

- Demonstrate: GenAI safe deployment concepts applied to design, development, TCQM and deployment in Safety Critical Product Lifecycle (PLC)
- Tactical Objective: Achieve Safe AI-augmented review of SRS
- Strategic Objective: Introduce AI Generation with Human-over-AI-over-AI governance

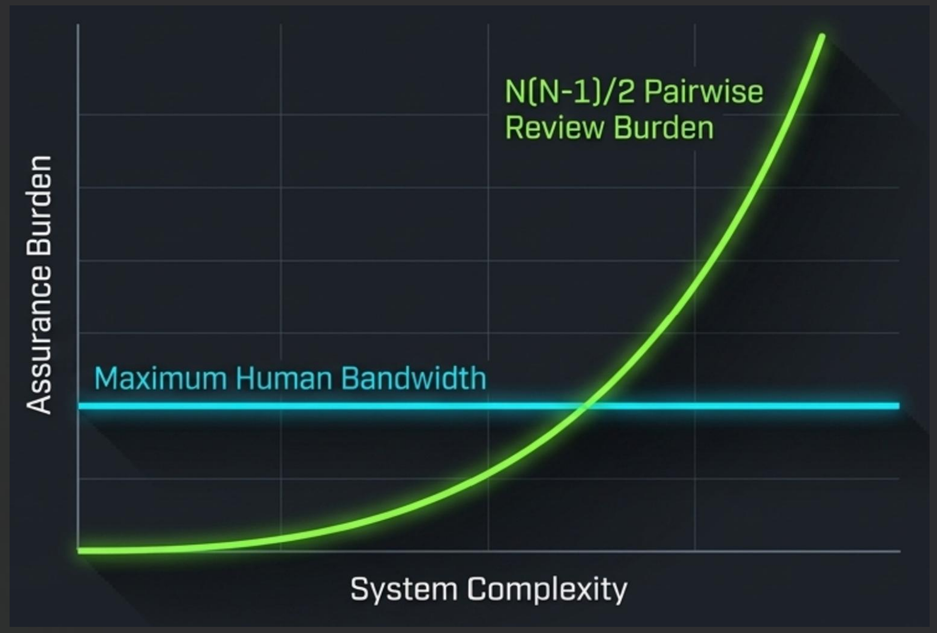
- **SRS** = Software Requirements Specification
- **TCQM** = Tool Classification, Qualification, Management

The compliance gap

Manual compliance does not scale; neither does optimism.

Artifacts volume and cross-domain coupling mathematically outrun human bandwidth

…“probably correct” is a very expensive phrase…

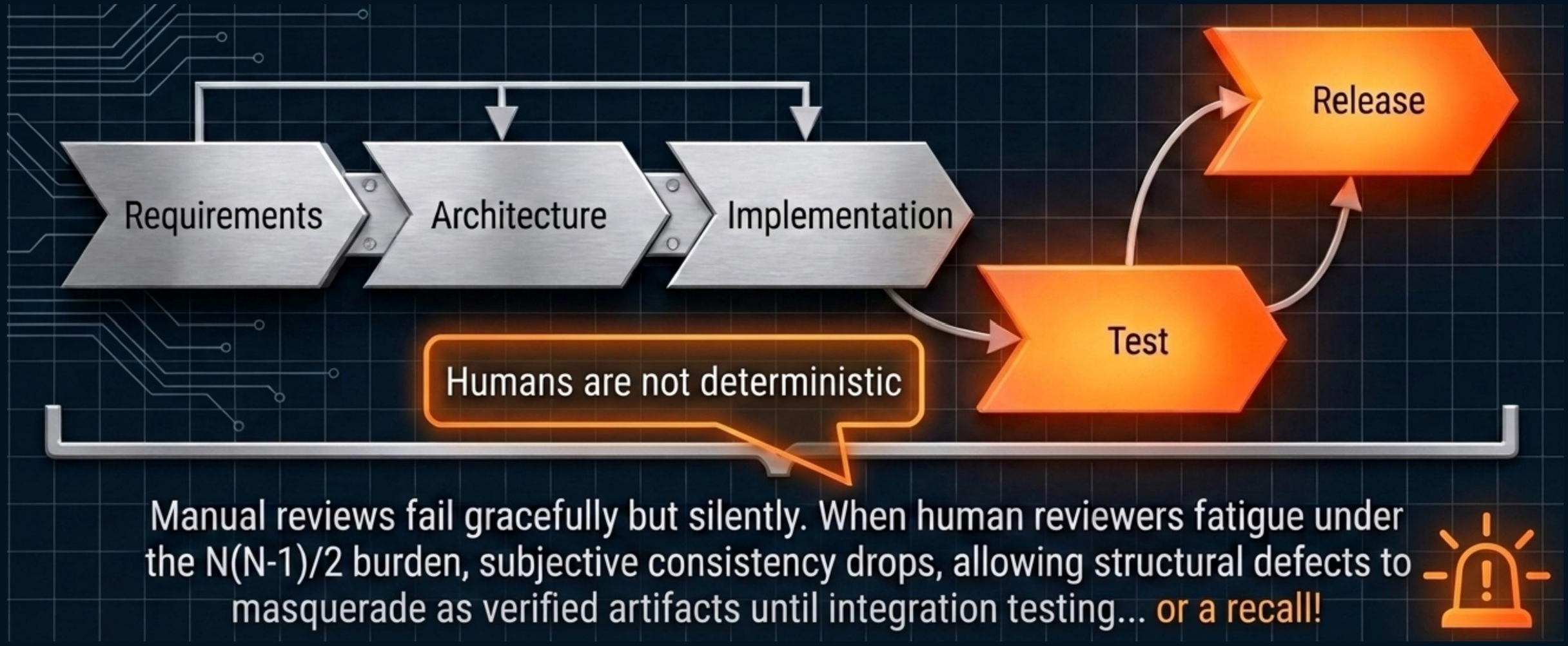


- Thousands of artifacts → combinatorial explosion of review pairs: 10s of Millions
- Manual review alone → tens of thousands of engineering days, non-scalable.
- Conclusion: automation is a compliance necessity, not a luxury.

- In this slides we refer to Artifacts or Work Products to identify data produced during the PLC

Lifecycle phases share one explicitly dangerous risk

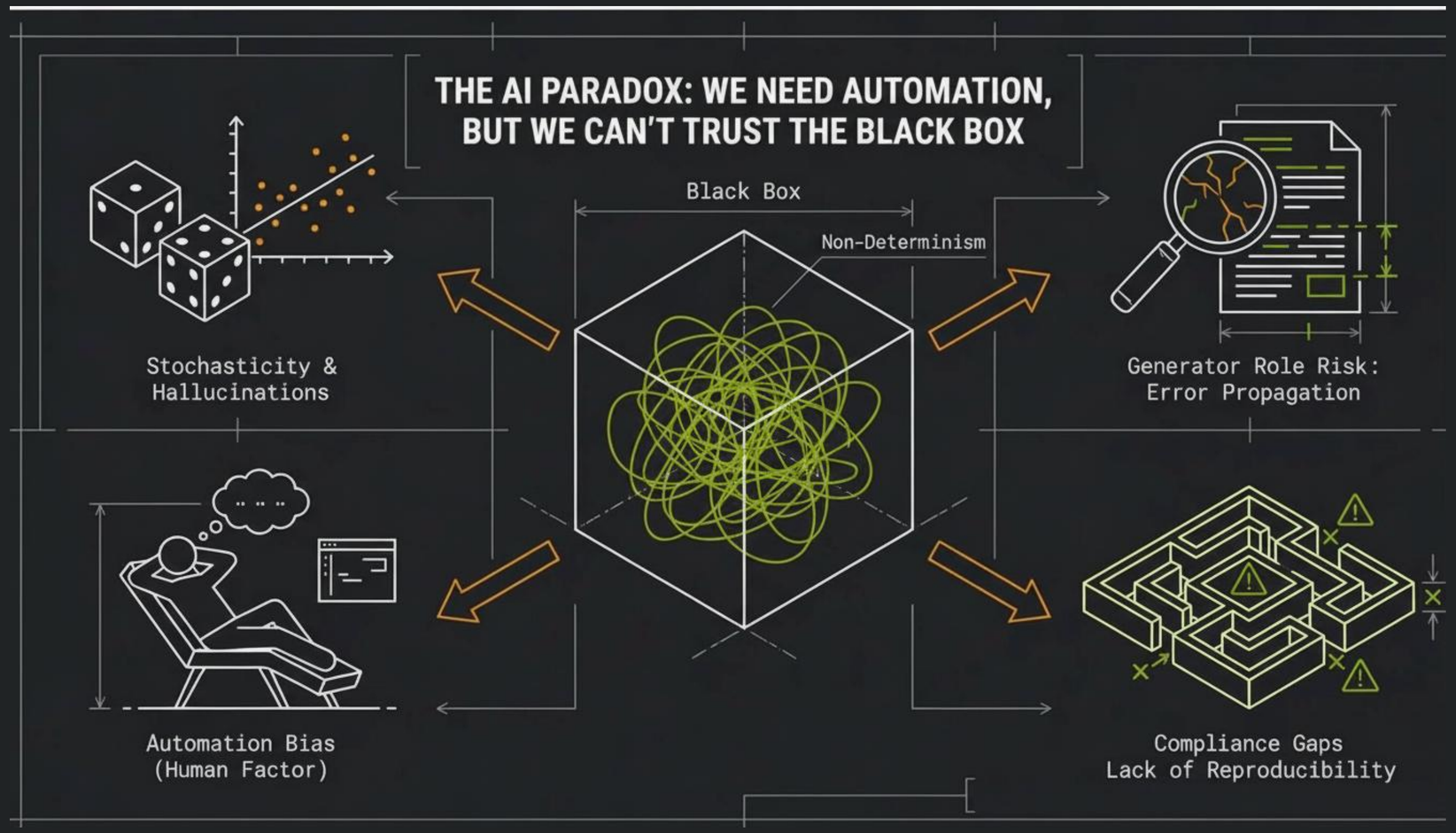
Late Discovery



The fundamental conflict between AI and Functional Safety

We need automation but can't trust black box, We want determinism but must leverage probabilism

- AI Autonomy
  - Observability
  - Interpretability
  - Inherently Probabilistic
- Reviewer Risk
  - Automation Bias
  - False Negatives leading to Complacency
  - Overconfidence



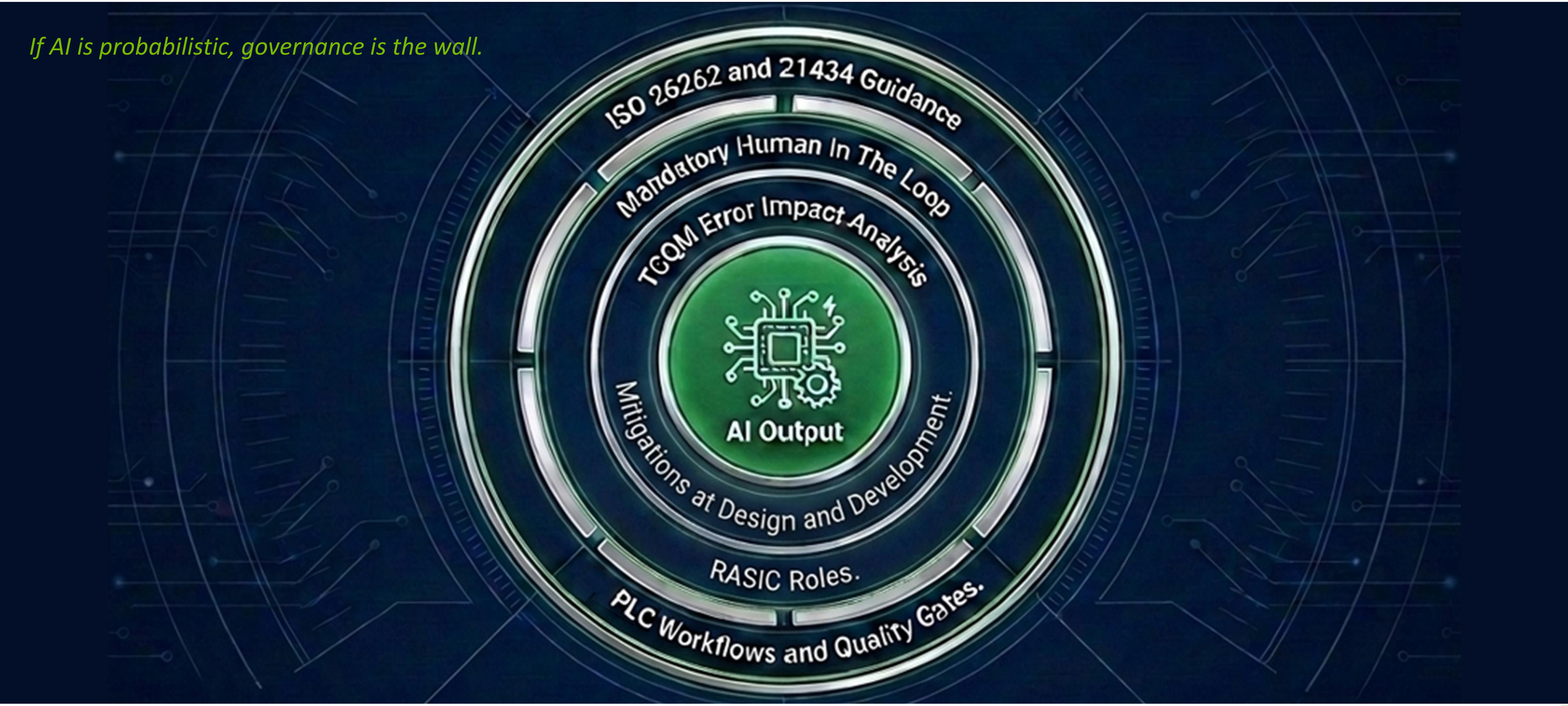
- Generator Risk
  - Autonomous Editing of Work Products
- Compliance Gaps
  - Lack of Reproducibility Conflicts with ISO 26262 expectations for safety-critical activities

We must constrain the AI role and architecture to restore **system-level determinism**.

The Fortress Metaphor

Wrapping probabilistic AI in deterministic process: Governance

If AI is probabilistic, governance is the wall.



Probability hurting, controls containing

Failure Mode	Mechanism	Control
Confident Hallucination	Overfitting to style over substance	Judge-agent cross-check + strict citation policy to knowledge base.
Silent Omission	Low recall on rare hazard cases	Tuned for Recall over Precision; Red-teamed Golden Sets of edge cases.
Automation Complacency	Humans rubber-stamp AI output (Automation Bias)	Mandatory sampling audits; secondary peer review on AI-approved items.
Scope Creep	Helpful edits outside defined intent	Hard prompt boundaries + diff-only allowlists.

The Reviewer Principle

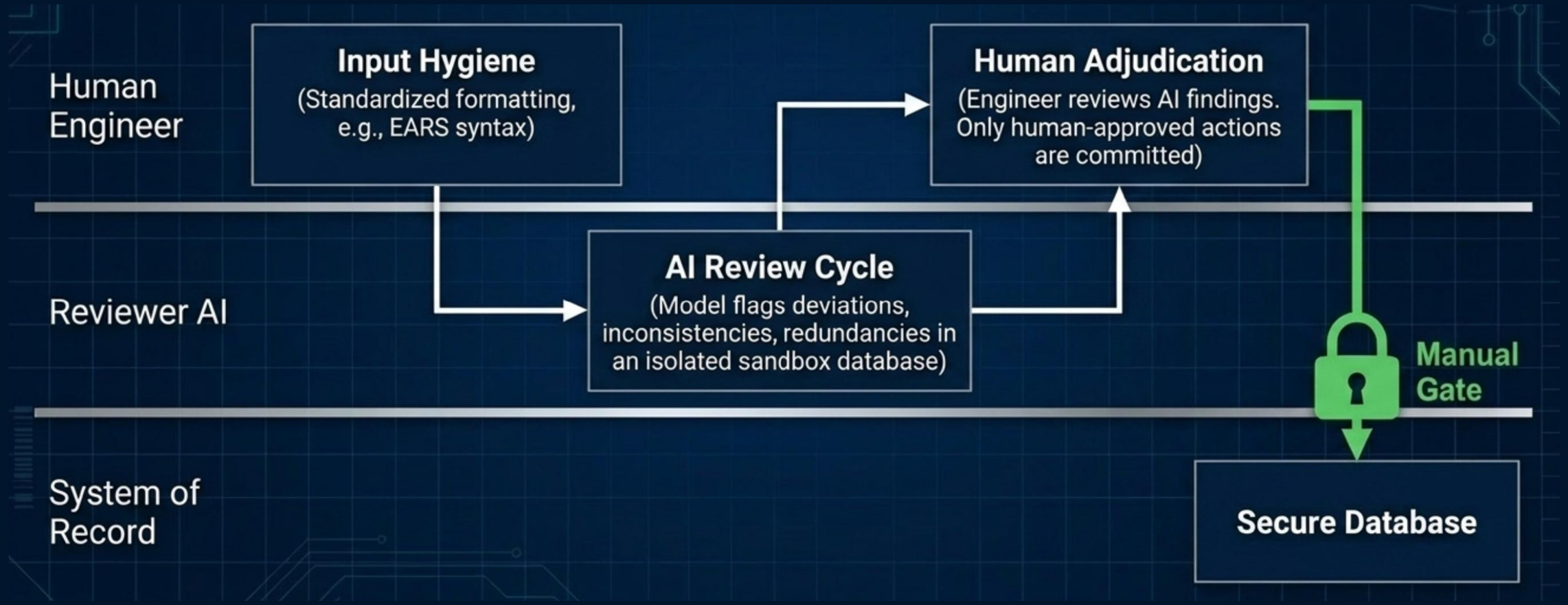
Deployment Reality

Human decides release content	Approval gates (RASIC) remain unchanged; AI outputs are strictly non-authoritative.
Zero autonomous change	No direct merge/write access to the System of Record (Jama, Git) without a human-in-the-loop process gate.
Evidence over fluency	Every AI-assisted decision generates an auditable Evidence Pack (metadata, hashes, rationales).
High recall for hazards	Reviewer models are tuned aggressively to catch safety defects, accepting higher false-positive nuisance.
Observable automation	Continuous background logging, sampling audits, and drift monitoring.

Let AI raise the flag, not rewrite the law.

Reviewer modules must land first

The adoption path



Helpful, review-oriented, and nowhere near the save button. Not yet.

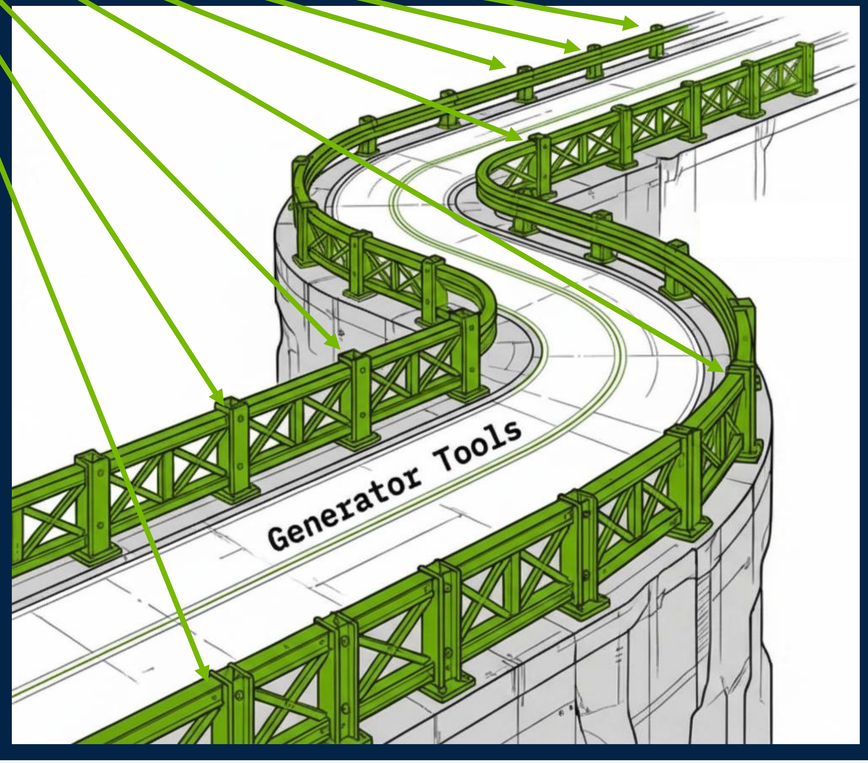
The First Deployment Principle: AI as Reviewer, not Creator

Safe pattern to Generators

vs.

Humans stay in the loop, but we are giving the loop better eyesight.

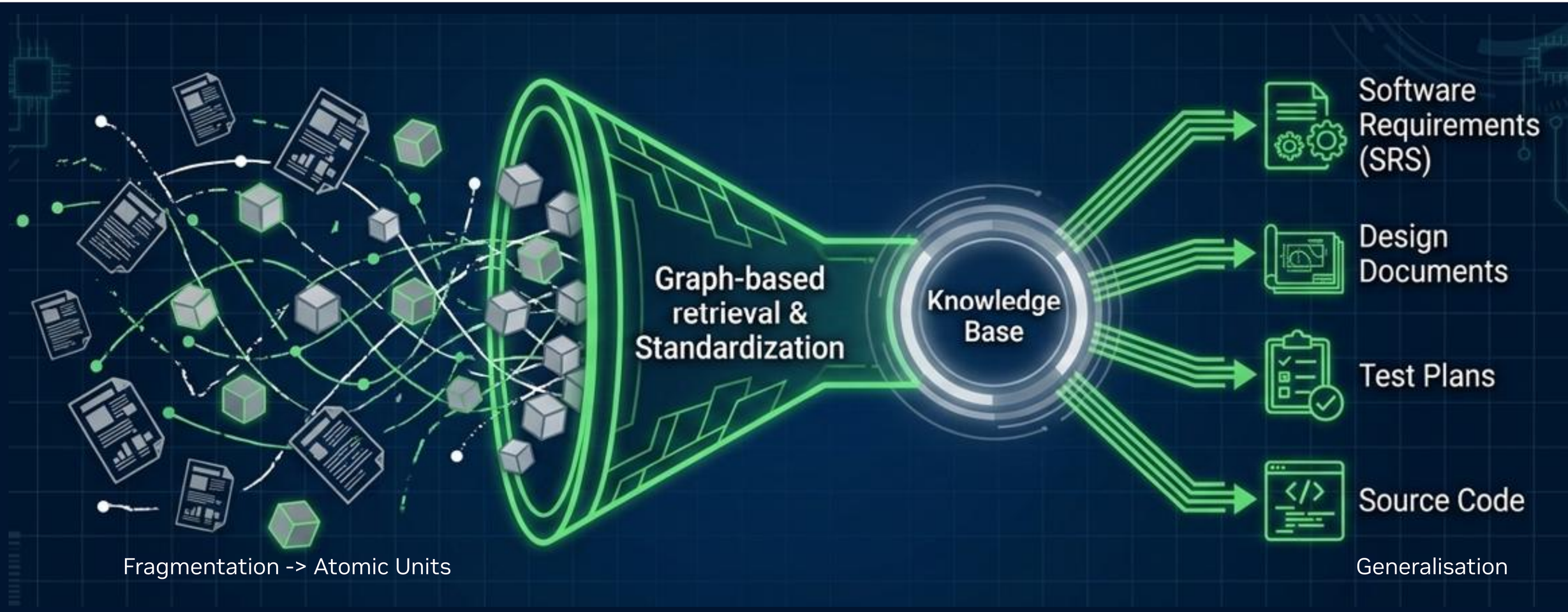
Safe path of Generators



- Why Reviewers
  - Gatekeeper Function: Reviewers Check Human Work
  - AI grades, does no editing; human decides
  - Risk limited to “Overconfidence” vs “Work Product Error Generation”
  - The **Guardrail Effect**: A solid Reviewer Workflow today becomes safety net for Generation Workflow tomorrow > **Human-over-AI-over-AI governance**

Standardized Input hygiene Enables Generalization

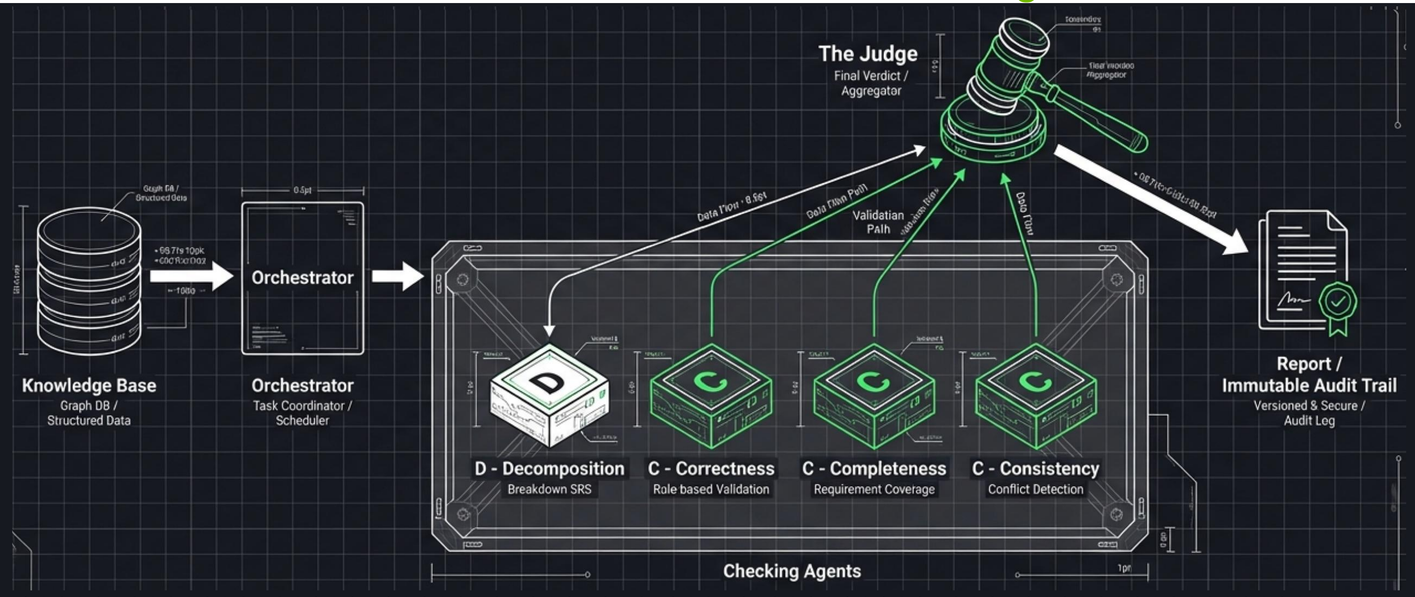
Support Re-use and Stabilization

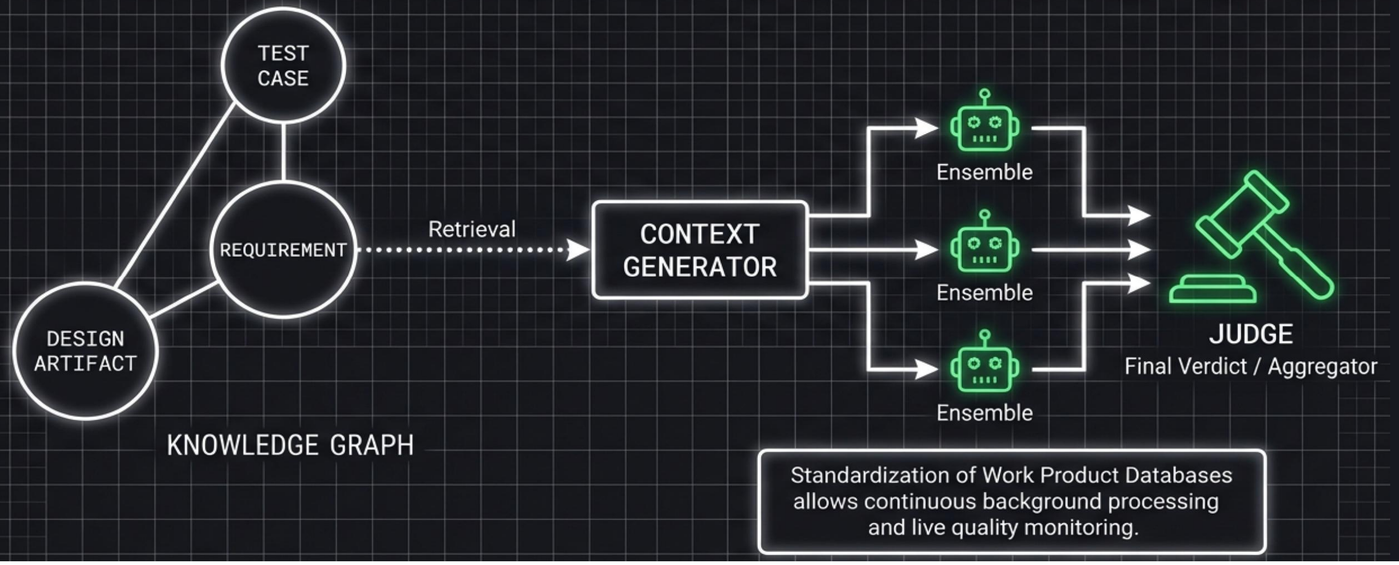


- Standardized artifact format and KB onboarding reduce input noise.
- Graph-based retrieval ensures consistent, versioned input for AI, improving stability.
- Re-use and Generalisation of interfaces and modules enable scalable “AI-Reviewer gated system”.

Specialise, Repeat, Combine, Expand

Building Blocks for Generalisation





- DCCC Verification Concept applies to any Set of Work Products:
  - Deconstruct -> KB atomic data
  - Check -> Decomposition, Correctness, Completeness, Consistency
  - Verify -> Analytics DB Records
- Knowledge Base across Work Products:
  - Standard Input
  - Granularity
  - Traceability
- Expand the DCCC-SRS Concept to other Work Products
- Leverage Context
- Leverage Ensemble


FALSE POSITIVE
OK
PROBLEM
SECURITY


FALSE NEGATIVE
OK
PROBLEM
SECURITY

Safety trade-off: Recall vs Precision & the cost of safety

Prefer false positives over false confidence

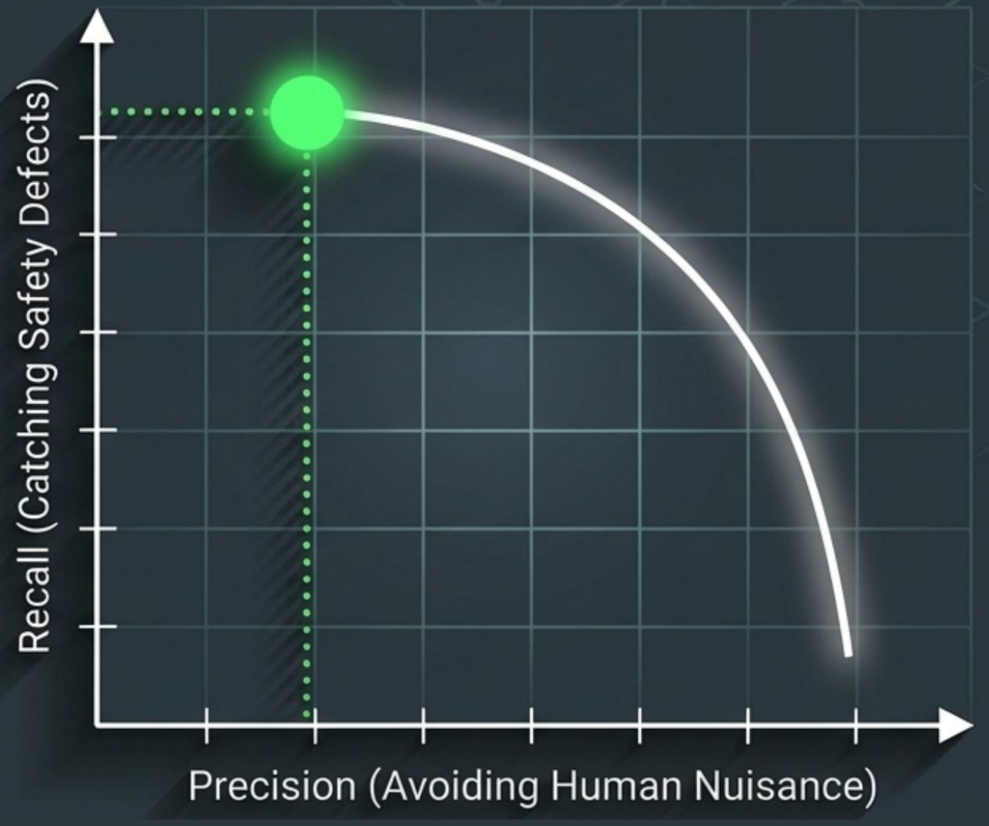
The Rule

In functional safety, we heavily penalize **Silent Omissions** (**False Negatives**). High recall minimizes missed hazards.

The Reality

We explicitly accept a higher **Nuisance Rate** (**False Positives**). One redundant manual review triggered by an overzealous AI is incredibly cheap compared to a masked defect escaping to the integration phase.

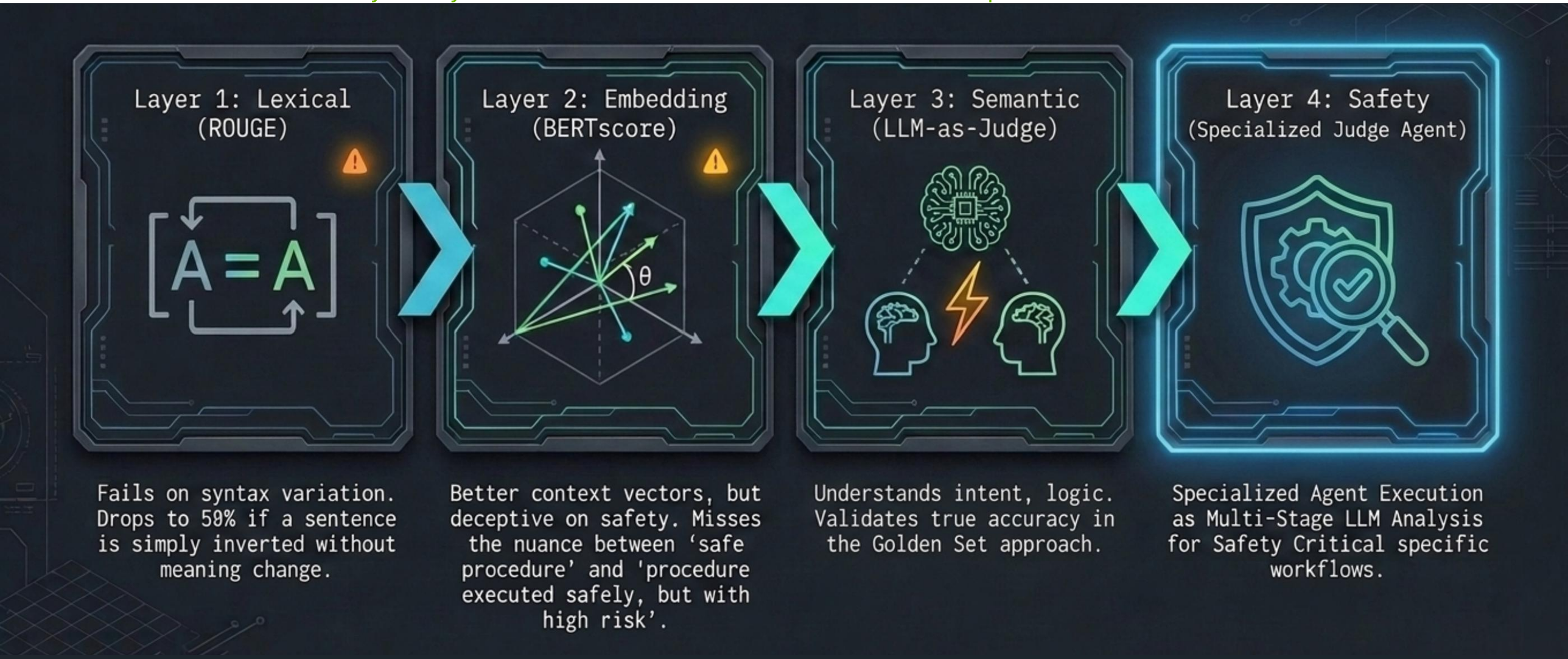
Nuisance is cheaper than silence.
Better an annoying alert than an invisible defect.



- Tune Reviewers for high Recall to minimize False Negatives (a.k.a. missed defects / masked errors).
- Mitigate False Positive Nuisance: robust model selection; ensemble voting; contextual embedding strategies; dimensionality and compression analysis; prompt tuning
- Argument to accept Residual Nuisance: one more review is cheap if truly a repetition, is beneficial if not yet reviewed

Golden Data Set: ROUGE, BERT and the JUDGE

Beyond syntax and Words semantic: metrics that keep humans in control



Traditional metrics mis-score safety-critical nuances. A multilayered measurement stack is required.

Think TCQM-oriented Breakdown Early in Design

Evaluate risks in each Processing Phase

Phase 1 Service Initialization and Configuration
Deterministic

Phase 2 Task Request Reception and Validation
Deterministic

Phase 3 Knowledge Base (KB) Input Retrieval
Deterministic

Phase 4 Vector Store Preprocessing and Ingestion
Hybrid

Phase 5 Input Data Combiner
Hybrid

Phase 6 Specialized Agent Execution - Multi-Stage LLM Analysis
LLM-Dominated

Phase 7 Results Serialization and Storage
Deterministic

Phase 8 Analytics Database (ADB) Integration (Results & Reviews Records)
Deterministic

Deconstruct your design to analyse each single phase for residual error propagation risks: deterministic vs probabilistic.

Risk Analysis: Errors and Error Propagation Impact

Prevention, Detection, Mitigation

Risk	Processing Phase	Nature	Propagation	Impact	Defense
R1 Malformed Input from KB	3 (Input from KB)	Deterministic	False Positive/Negative Instances. Unintelligible content.	Corrupted requirements. Waste of resources.	Prevention involving User. Prevention by Input Tool. Detection in SRS-DCCC Tool
R2 Malformed Output in ADB	8 (Output to ADB)	Deterministic	Flipped flags result in FP/FN. Corrupted Results file.	User forced to abort review. Waste of resources.	Prevention involving User. Prevention at SRS-DCCC/ADB Interface. Detection in ADB Tool
R3 False Positives	6 (LLM Agents)	Non-deterministic	Incorrect verdict reported in results.	Engineer misinterprets finding. Potential safety impact, rework cost.	Mitigation in Tool. Prevention (workflow guardrails). Detection (downstream activities)
R4 False Negatives	6 (LLM Agents)	Non-deterministic	False Negative masks SRS error.	Defect detected late in PLC. Rework cost, safety impact.	Mitigation in Tool. Prevention (workflow guardrails). Detection (downstream activities)

- ❑ Detailed Error Analysis for each Processing Phase
- ❑ Document Internal Errors that are detected/prevented systematically
- ❑ Perform Impact Analysis for Errors that can Propagate to the Output

If we cannot classify the risk, we should not deploy the magic.

Example: High-level Tool Classification Report

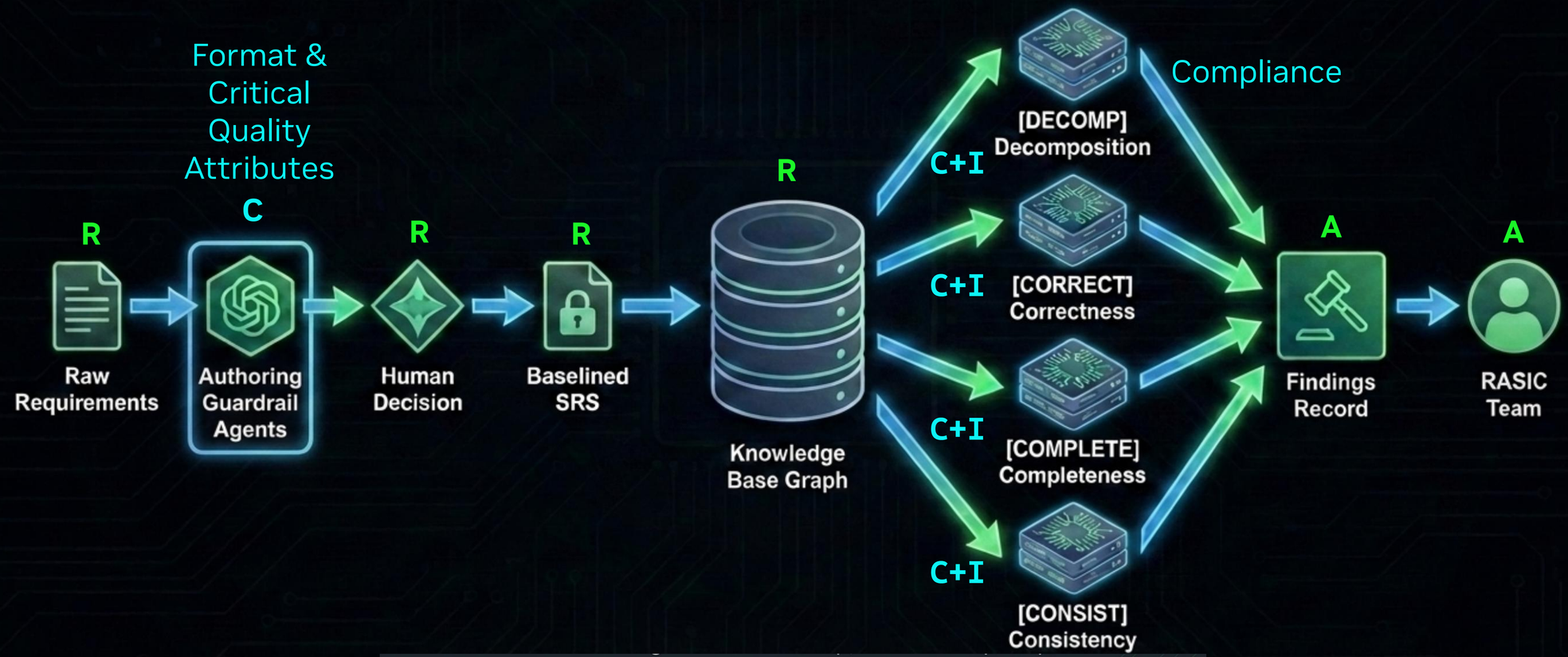
TCQM Positioning and Expectation

Item	Our Position	Notes
AI Tool Usage (AI-TU)	AI-TU1 (lower AI related risk due to indirect impact on Safety)	AI stages of DCCC-SRS cannot modify SRS autonomously. All errors and risks analysed.
Tool Impact	TI2 (risk to introduce or mask errors)	DCCC-SRS can influence safety-related SRS but does not modify them autonomously. All risks assessed.
Tool Error Detection (TD)	TD1 (high degree of confidence)	Blend of: Mitigation by design/development, Detection of input/output errors, Prevention by workflow guardrails, Detection by downstream activities

- **Expected TCR Assessment → TCL1** — DCCC-SRS is classified as an assistance tool; it must not be used as sole evidence of SRS adequacy.
- **Evidence Contribution:** DCCC-SRS findings can contribute to quality evidence as supporting evidence.
- **Argumentation:** ‘Requirements quality validated through human review augmented by DCCC-SRS automated analysis.’

End-to-End Requirements Evaluation to Cross-Artifact Verification

Safe AI scales through clear ownership, not ad-hoc prompt culture

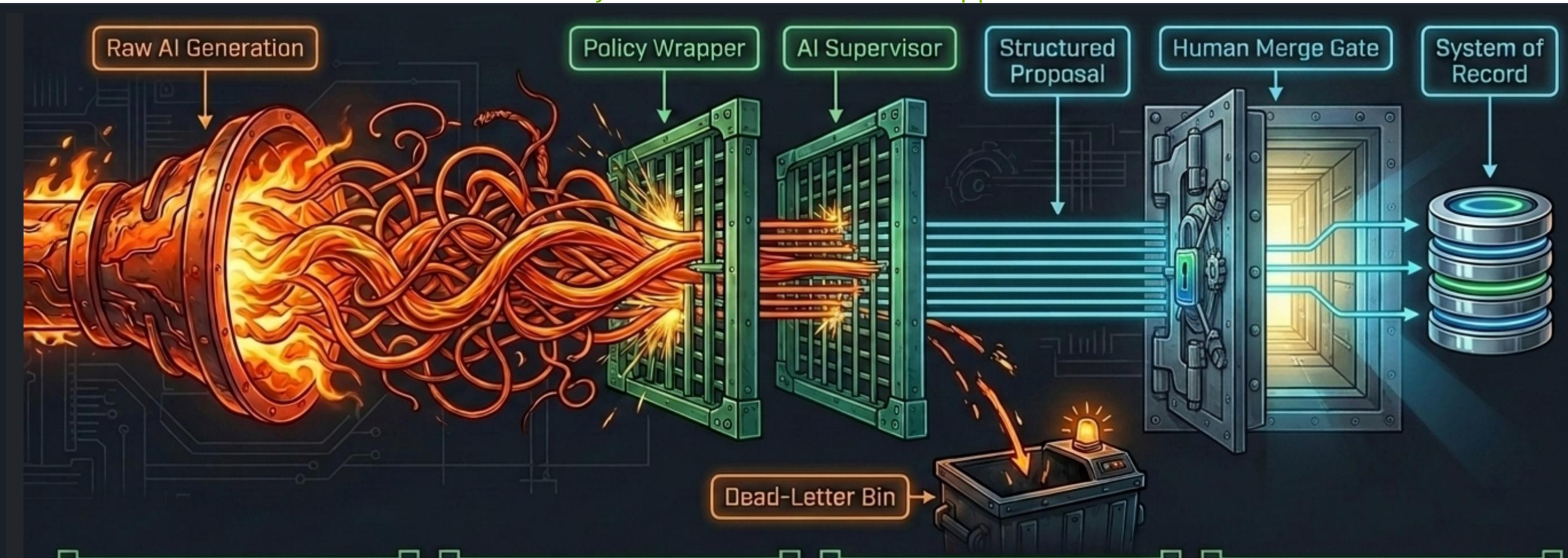


Task	Human Author	Reviewer Model	Supervisor Model	Safety/Security PIC
Input Definition	R			
Structural Check		C		
Policy Check			I	
Final Authorization				A

We do use AI creatively — just not where it can creatively rewrite the requirement.

Generation Workflow

Only Behind Deterministic Wrappers



Input Hygiene:
Strict standardized templates limit context windows.

Enforced Determinism:
Temperature set to `0.1`; rigid system prompting.

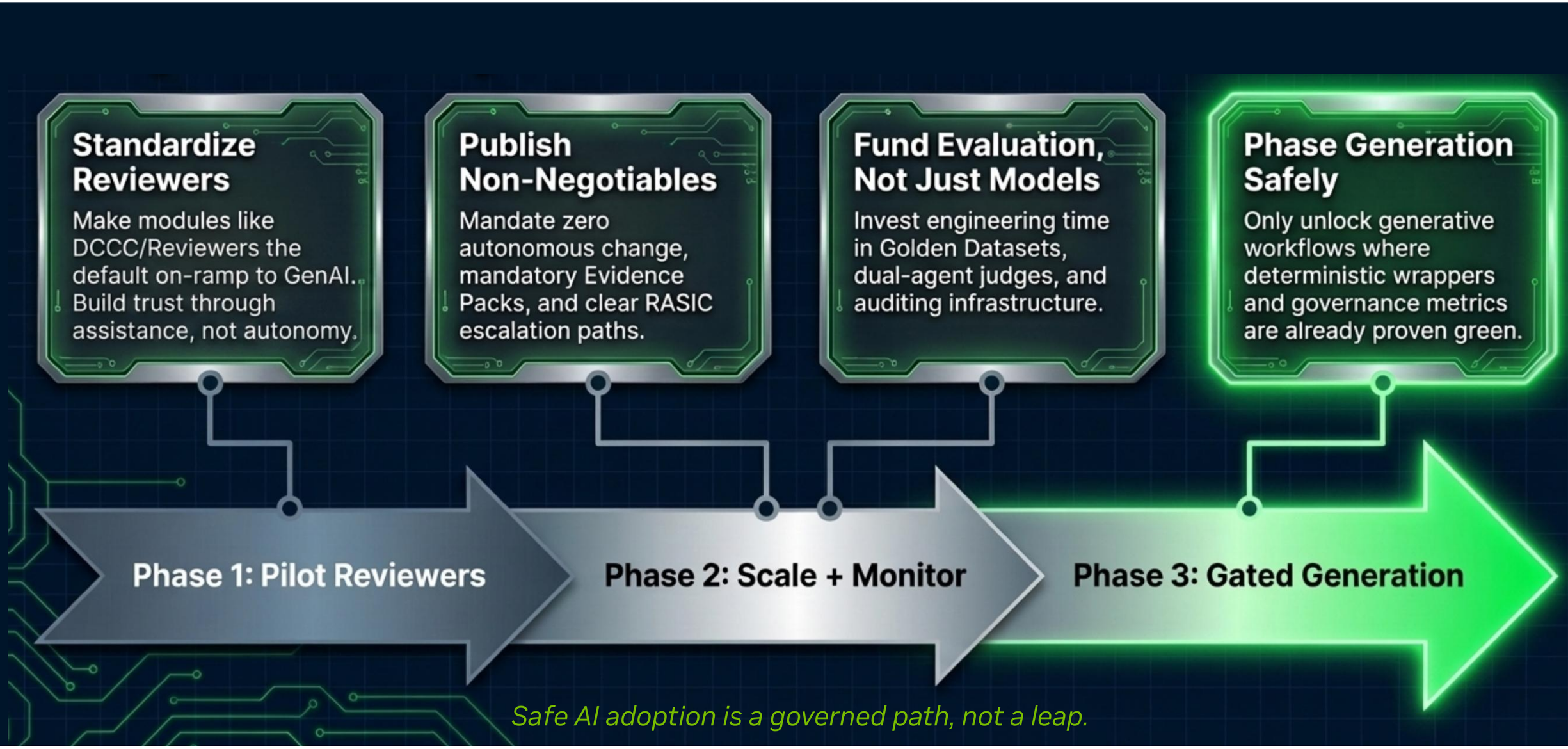
Diff-Only Proposals:
Generation never rewrites the baseline; it only proposes atomic diffs.

Mandatory Adjudication:
Content is dead-lettered unless explicitly merged by a human.

Creation is earned after review becomes boringly reliable.

Phasing Generative AI Adoption Safely

Invest in Reviewers & Governance



Safe AI adoption is a governed path, not a leap.

Thank you  for the opportunity.

To get involved reach out to: Eli egurvitz@nvidia.com