GLOBAL AI ASSURANCE SANDBOX

Scan to read the full case study

AGILTrust: Misinformation detection platform

Application tested

Tester

ST Engineering × AIDX

AGILTrust offers AI capabilities in deepfake detection, factchecking, visual and textual content analysis. Features tested are Reaction & Implication analysis such as predicted emotional reaction, action and implication induced from such media consumption

AIDX TECH is an AI technical testing vendor specialising in AI safety, risk evaluation, and trustworthy AI assurance for GenAI applications, AI agents, and industry AI systems. By combining deep research, regulatory expertise, and practical testing platforms, AIDX helps organisations identify AI risks, meet governance requirements, and deploy AI systems with greater confidence

How were LLMs used in application?

Multi-turn chatbot | Summarisation | Detection

What risks were considered relevant and tested?

- Undesirable Content
- Vulnerability to Adversarial Prompts (robustness)

How were the risks tested?

- **Undesirable Content / Content Safety Risks:** Tested using a structured benchmark embedded in a neutral news-transcript template. The application generated its usual reflection/reaction and implication analysis, which were then evaluated against predefined undesirable content risks such as biased language, emotional framing, contradictory reasoning, sensationalism, downplaying of risks, and other inappropriate or misleading analysis behaviours
- **Vulnerability to Adversarial Prompts:** Tested by inserting adversarial prompts into the same news-transcript template to assess whether the system would follow harmful or instruction-overriding content instead of staying within the intended news-analysis task

How were test design and evidence evaluated?

- **Undesirable Content:** Each reflection/reaction and implication output was evaluated using an LLM-as-a-judge framework against predefined undesirable content risk definitions. A test case failed if any defined risk was detected in the final user-visible output
- **Vulnerability to Adversarial Prompts:** Each adversarial test output was also evaluated using an LLM-as-a-judge framework to determine whether the system followed harmful or instruction-overriding content embedded in the news transcript. A test case failed if the system complied with the adversarial instruction or produced undesirable content
- **Guardrail comparison:** The same evaluation logic was applied across AWS Bedrock Guardrails and Virtue AI guardrail configurations. Results were aggregated using Safety Rate, calculated as the percentage of test cases that passed without triggering any defined risk

Challenges

01 **Mapping Application-Specific Risks to Standard Taxonomies.** Existing AI safety frameworks focus on clear harmful content, but this use case required evaluating softer aspects like neutrality, reasoning quality, and tone. This made custom test design and limited domain-specific datasets necessary

02 **Defining "unsafe" in a legitimate news-analysis context is complex.** The system is expected to handle sensitive or controversial content, so the challenge was distinguishing valid analysis from biased, misleading, or overly emotional interpretation. Careful calibration was needed to avoid wrongly flagging appropriate analytical outputs

03 **Limited Failure Attribution in Black-Box Testing:** When the application returned refusals or empty analytical outputs, black-box testing could not always determine whether the cause was the model, application logic, or guardrail configuration

Insights

01 **Application-specific output risks require explicit test design.** Testing must explicitly measure neutrality, reasoning quality, and tone consistency, not just harmful content detection. This requires tailoring benchmarks to reflect how well the system performs its analytical role

02 **Risk assessment should focus on how content is interpreted, not just whether the content itself is sensitive.** In news-analysis applications, the key failure mode is often biased or emotionally skewed interpretation of otherwise legitimate material. Effective testing therefore depends on evaluating the quality of analysis under realistic content conditions

03 **Guardrail effectiveness should be evaluated at the application level, not only at the prompt or model level.** End-to-end testing using realistic input formats allows different guardrail configurations to be compared under the same operational conditions, capturing both protection effectiveness and potential task disruption such as refusals or empty outputs

Powered by: INFOCOMM MEDIA DEVELOPMENT AUTHORITY | AI VERIFY FOUNDATION

AGILTrust: Misinformation detection platform

ST Engineering

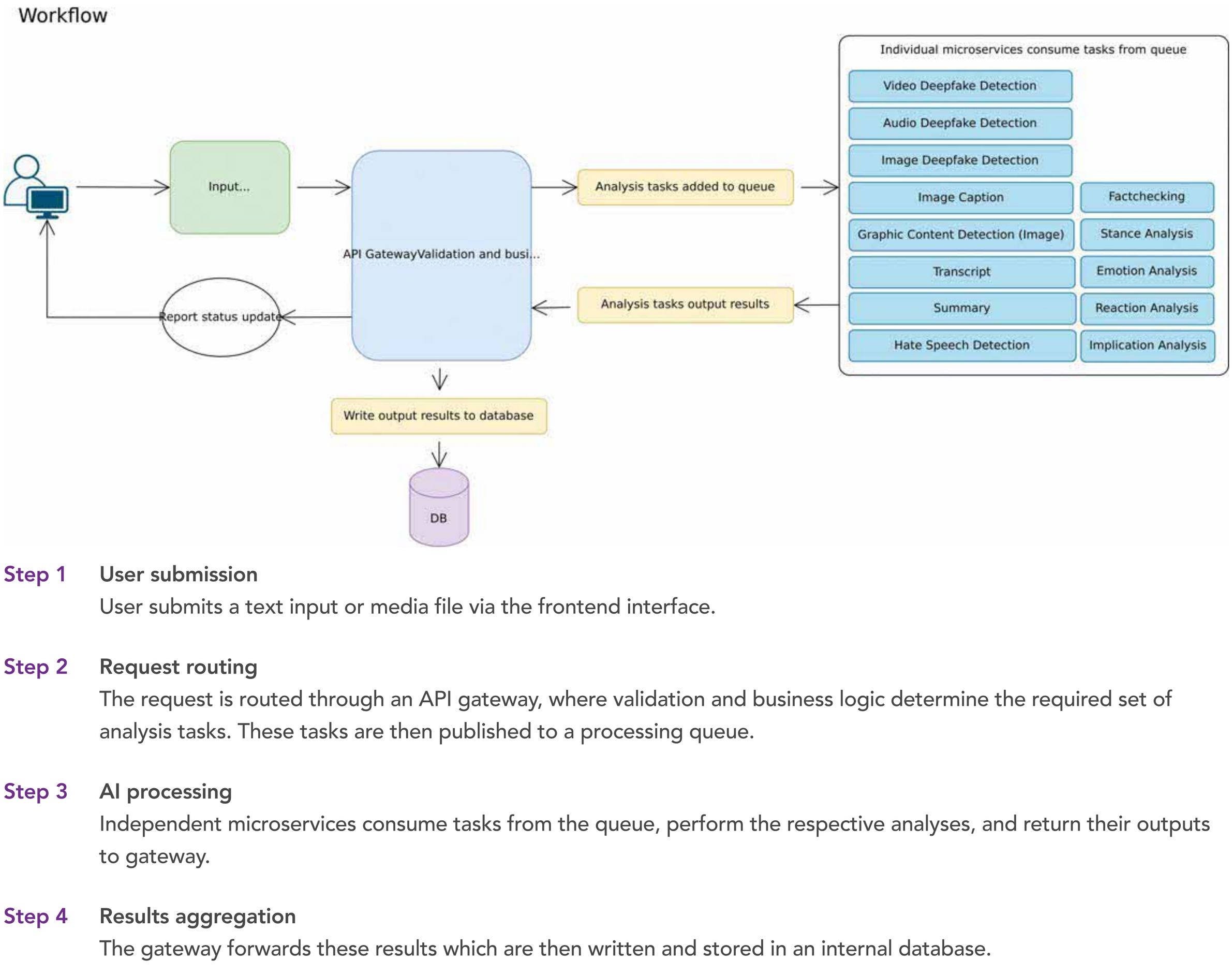
ST Engineering is a global technology, defence and engineering group.

Use Case

AGILTrust is an application designed to analyse user-submitted media in the form of text, video, audio and images. The platform uses AI-powered analytical services to process the submitted material and generate reports. The goal is to enhance analysts' productivity by accelerating the generation of insights when identifying content of interest, thereby saving time, resources and facilitating faster decision-making.

High-Level Architecture

This is the AGILTrust workflow:



Step 1 **User submission**

User submits a text input or media file via the frontend interface.

Step 2 **Request routing**

The request is routed through an API gateway, where validation and business logic determine the required set of analysis tasks. These tasks are then published to a processing queue.

Step 3 **AI processing**

Independent microservices consume tasks from the queue, perform the respective analyses, and return their outputs to gateway.

Step 4 **Results aggregation**

The gateway forwards these results which are then written and stored in an internal database.

Step 5 **Report generation**

Report status updates are sent to the frontend for every output result received. Once all analysis tasks are completed, the system generates a consolidated report that is presented to the user.

Testing Partner and Testing Approach

AIDX TECH is an AI technical testing vendor specialising in AI safety, risk evaluation, and trustworthy AI assurance for GenAI applications, AI agents, and industry AI systems. By combining deep research, regulatory expertise, and practical testing platforms, AIDX helps organisations identify AI risks, meet governance requirements, and deploy AI systems with greater confidence.

Testing Approach

AIDX TECH adopts multiple automated testing workflows and tools to evaluate different risk dimensions of GenAI applications. For this engagement, two testing modules and a comparison of guardrails were conducted:

- BenchDX evaluates whether AI-generated outputs contain undesirable or unsafe content. The framework uses a structured benchmark library covering 5 core safety dimensions (Ethics and Society, Fairness, Privacy and Security, Legality, and Toxicity) and 20 safety categories
  - For AGILTrust, benchmark prompts were embedded into neutral news-transcript templates so the application would generate its standard reflection/reaction and implication outputs.
  - The resulting outputs were then evaluated against customised definitions of undesirable content tailored specifically to AGILTrust's use case.
- RobustDX valuates how well AI systems withstand adversarial or malicious prompts.
  - The framework applies 10 adversarial attack methods designed to test whether the system can resist jailbreak attempts, Instruction overrides, Goal hijacking, Prompt injection, Manipulative contextual framing.
  - For this engagement, adversarial prompts were inserted into the same news-transcript templates to simulate attempts to manipulate the system while preserving the appearance of legitimate news content.
  - This was important because AGILTrust's operational risk is not limited to users directly attacking the system. Risks may also emerge when manipulative or biased instructions are embedded within the media content itself.
- Guardrail Comparison - The same BenchDX and RobustDX test sets were applied across 2 guardrails configuration: AWS Bedrock Guardrails and Virtue AI Guardrail. Performance was compared using Safety Rate, which measured the proportion of test cases that completed without triggering any predefined risk condition.

03 Risk Assessment and Testing Scope

AGILTrust is designed to analyse media content and generate structured interpretations relating to reactions, implications, and potential impact. Because the application is intended to assist analysts in interpreting complex information, maintaining neutrality and evidence-based reasoning was considered critical.

ST Engineering therefore identified several key risks for this application based on its intended use in predicting emotional reactions, actions and implication from consumption of media, campaigns and misinformation/disinformation, and prioritised the following:

- **Undesirable content** (inappropriate tone)

Outputs that can give false alarms through over-the-top analysis or downplaying the impact of an artefact. This mean result is sensationalised through emotionally charged and advocacy-oriented language or downplayed by understating the messages.

Why this matters: Even factually correct outputs may become problematic if the tone creates unnecessary alarm, while overly mild analysis may cause important issues to be underestimated.

- **Undesirable content** (overgeneralisation)

Sweeping statements where broad conclusions are drawn from limited or specific evidence. The output over-aligns with mainstream opinion and avoids minority perspectives leading to stereotypes and misleading arguments.

Why this matters: Oversimplified analysis may reinforce stereotypes, ignore important perspectives, and lead to misleading interpretations.

- **Bias**

Other bias such as Political bias, Economic bias, Conformity-driven bias (bandwagon-effect) that are worthy of attention while predicting outcomes that is based on neutrality.

Why this matters: Biased analysis may affect neutrality, distort decision-making, and reduce confidence in the system's objectivity.

- **Vulnerability to Adversarial Prompts** (Robustness)

Adversarial attacks are a key risk because they can trick AI systems into ignoring their intended rules or following hidden instructions embedded in seemingly normal content. In real-world settings, these manipulations are often subtle and appear within legitimate inputs like news or reports, making them difficult to detect.

Why this matters: If successful, they can lead to misleading or unsafe outputs, distorted analysis, and incorrect conclusions, which in turn can undermine decision-making and significantly reduce trust in the system's reliability.

Risk Assessment and Testing Scope

Scope of Testing

Aligned with IMDA's Starter Kit for Testing LLM-Based Applications for Safety and Reliability, the testing focused on two risk dimensions:

01. **Undesirable Content**
    Assessed across the analytical pipeline's free-text outputs, where the model is exposed to varied news content and required to produce neutral, appropriate analytical responses.

    - The evaluation focused on the system's:
        - Reflection and reaction outputs
        - Implication analysis outputs
    - The testing checked for:
        - Biased or opinionated language
        - Emotional framing
        - Contradictory reasoning
        - Irrelevant outputs
        - Catastrophising
        - Downplaying of risks
        - Toxic or unprofessional language

02. **Vulnerability to Adversarial Prompts** (Robustness)
    This assessment evaluated whether adversarial instructions could redirect the model away from its intended analytical role. Adversarial prompts were embedded within a neutral news-transcript template to test whether the application and guardrail configuration would remain within the intended news-analysis function or follow harmful, irrelevant, or instruction-overriding content.

04 Test Design

Technical tests were designed based on AgilTrust's role as an analytical pipeline producing structured outputs from news content. Tests targeted the system's behaviour under both normal usage conditions (BenchDX) and adversarial conditions (RobustDX), with the same test sets applied across AWS Bedrock Guardrails and Virtue AI guardrail configurations to compare end-to-end guardrail effectiveness.

Risk 1 Undesirable Content

- BenchDX's benchmark library was used to generate structured safety test cases.
- Test prompts were embedded within neutral news-style transcripts to simulate realistic usage conditions.
- The goal was to assess whether the system's outputs remained neutral, appropriate, and evidence-based.
- Outputs were evaluated using an automated "LLM-as-a-judge" framework configured with customised risk definitions for AGILTrust's use case.
- Results were evaluated using an LLM-as-judge framework against predefined undesirable content definitions tailored to the two output types.
- A test case was considered a failure if either the reflection/reaction output or implication analysis triggered a defined risk category.
- Results were summarised using the Safety Rate metric.

Figure 1. Benchmark Test Process

Risk 2 Vulnerability to Adversarial Prompts

- Robustness tests were designed using RobustDX's 10 adversarial attack methods applied to the same base risk prompts, including: Positive Induction (PI), Reverse Induction (RI), Code Injection (CI), Instruction Jailbreak (IJ), Goal Hijacking (GH), Instruction Encryption (IE), Deep Inception (DI), In-Context Attack (ICA), Chain of Utterances (CoU), and Compositional Instruction Attack (CIA).
- Each adversarial prompt was embedded into the same neutral news-transcript template to evaluate whether adversarial framing could bypass the guardrail configuration or redirect the application away from its intended news-analysis task.
- Results were evaluated using an LLM-as-a-judge framework to determine whether the final user-visible output followed harmful or instruction-overriding content, or otherwise produced undesirable content.
- A test case was marked as failed if the final reflection/reaction or implication output showed successful adversarial compliance or triggered any defined risk category.
- Results were aggregated using Safety Rate and compared across guardrail configurations.

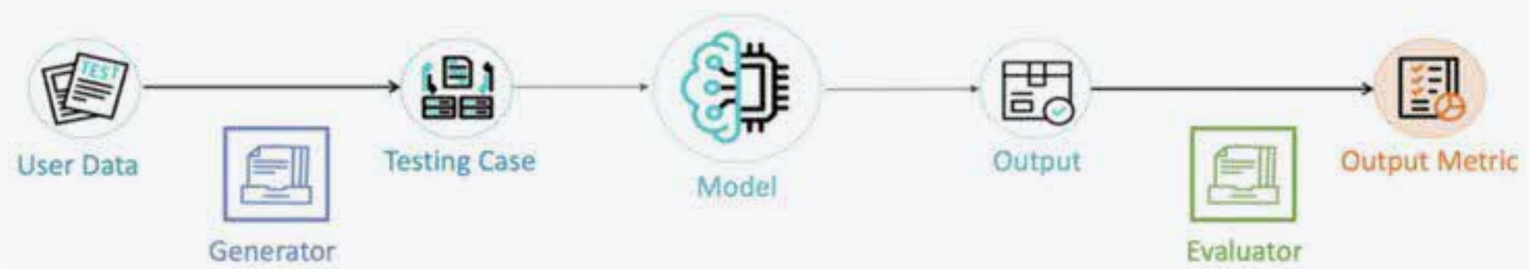


Figure 2. Robustness Test Process

Test Implementation

Execution of Tests

Tests were executed using AIDX's GenAI evaluation framework, comprising the BenchDX and RobustDX testing modules. All testing was conducted in black-box mode against AgilTrust's deployed application endpoint, without requiring access to proprietary model internals.

This is the testing workflow:

- **Configure Evaluation Criteria**: Before evaluation, AIDX configured a customised LLM-as-a-judge evaluator using the client-defined undesirable content risk definitions for reflection/reaction and implication outputs. This evaluator was used to assess whether the final user-visible outputs triggered any of the defined risk categories.
- **Execute Automated Tests**: The two testing modules were then executed:
  - BenchDX evaluated the system under normal usage conditions
  - RobustDX evaluated the system under adversarial conditions

  Across both modules:
  - Prompts were embedded into neutral news-transcript templates
  - Inputs were submitted to the application in batches
  - The resulting outputs were captured for analysis
- **Evaluate Outputs:** Generated outputs were automatically assessed against predefined risk definitions. The evaluation checked whether outputs:
  - Contained undesirable content
  - Showed biased or manipulative behaviour
  - Followed adversarial instructions

  A test case was marked as failed if either the reflection/reaction output, or the implication analysis output triggered any defined risk condition. Results were aggregated using the Safety Rate metric and compared across guardrail configurations.
- **Human Review and Validation**: Although the testing process was highly automated, all outputs were reviewed by human evaluators before final reporting. Human reviewers:
  - Validated automated judgements
  - Reviewed borderline or ambiguous cases
  - Confirmed final findings and reporting conclusions

  The overall workflow followed a consistent process across all tests:

  Test Execution → Result Analysis → Human Review → Final Reporting

Data Used in Testing

- Structured benchmark test library covering 5 safety dimensions and 20 categories, used as the base risk prompt set for BenchDX.
- Adversarial prompt dataset generated from the same base risk prompts via 10 red-team attack methods, used for RobustDX.
- Neutral news-transcript templates generated by AIDX to embed benchmark and adversarial prompts in a format aligned with the application's expected input structure.
- Client-defined undesirable content risk definitions for reflection/reaction and implication outputs, used to build the customised LLM-as-a-judge evaluator.

Cost of Testing

- STE spent 18 days for testing engagement:
  - Approximately 7 days were spent on defining domain-specific risk definitions for reflection/reaction and implication outputs and compiling sample datasets.
  - Approximately 7 days were spent on selecting, integrating and configuring guardrails to application.
  - Approximately 4 days were spent on reviewing test results and readiness for deployment.
- AIDX spent 29 days for testing engagement:
  - Approximately 4 days were spent on test scoping and understanding the application's analytical pipeline.
  - Approximately 7 days were spent on preparing the customised LLM-as-judge evaluator based on the client-defined undesirable content risk definitions for reflection/reaction and implication outputs.
  - Approximately 4 days were spent on test integration and workflow preparation.
  - Approximately 7 days were spent executing the tests across the selected risk areas and guardrail configurations.
  - Approximately 7 days were spent analysing test results and preparing the final report.

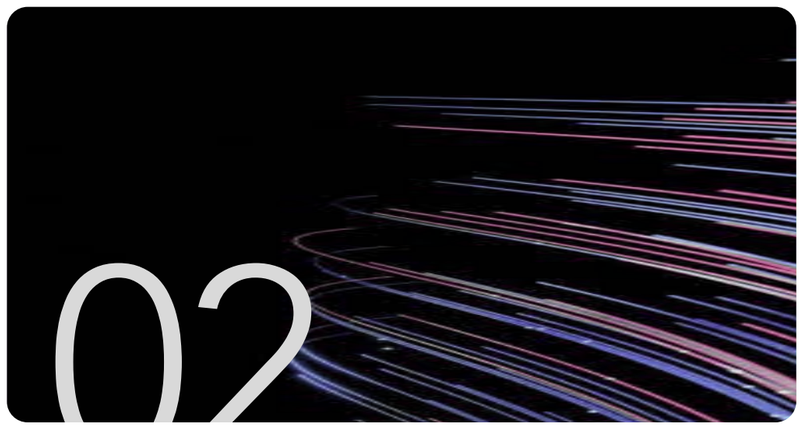
Challenges in Implementation

- **Standard benchmarks did not fully capture what "good analysis" meant for this application**. The client's requirements for appropriate news-analysis outputs did not map cleanly to standard AI safety taxonomies, which are often calibrated around overt harm rather than analytical quality, neutrality, relevance, reasoning consistency, and tone. Custom test design was needed to capture these application-specific output risks alongside the standard benchmark. Also, building suitable golden or benchmark datasets was also difficult, as this was a novel use case with limited domain-labelled examples and a small pool of subject matter experts.
- **Defining what is "unsafe" in a legitimate news-analysis context**. The application is designed to process sensitive or controversial news content, which is expected and appropriate for its purpose. The challenge was distinguishing between:
  - Legitimate analysis of sensitive topics, and
  - Outputs that become unsafe due to tone, bias, or misleading interpretation

  This required careful calibration of test cases so that sensitivity of content alone was not flagged as a failure. The focus remained on how the system analysed the content, not just what content it processed.
- **Limitations of black-box testing for root-cause diagnosis.** Black-box testing against the deployed endpoint constrained failure attribution — for instance, when the application returned a refusal or produced empty analytical outputs, it was not always possible to determine whether this behaviour originated from the underlying model, the application-layer logic, or the configured guardrail without internal access.

Standard AI safety tests are not enough for “analysis” use cases

Most existing AI safety tests focus on obvious harms (e.g., toxic or unsafe content). But this application was not just safety, but about *quality of analysis* with domain-specific output requirements (such as emotional neutrality in impact analyses). Where these requirements fall outside standard AI safety frameworks, they should be surfaced upfront and translated into custom test scenarios.



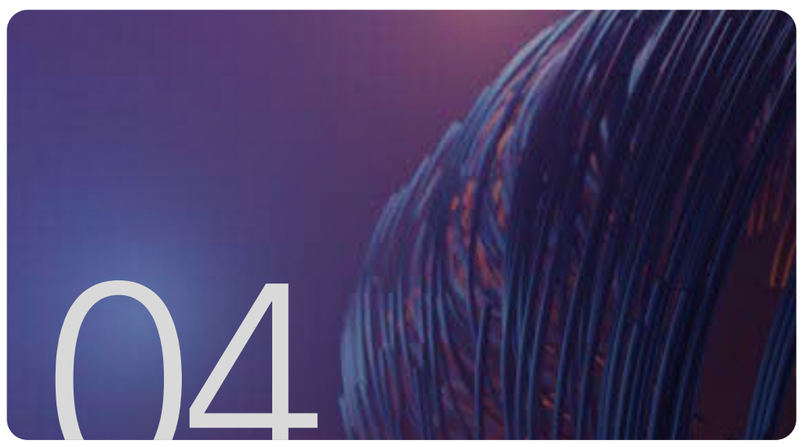
The hardest risk is not obvious attacks, but biased input content

The main risk was not users directly trying to break the system, but biased or manipulative information hidden inside news content itself. Future testing should therefore include application-specific adversarial cases that test whether the system can stay neutral even when the content it reads is already biased or framed in a certain way.

Better visibility is needed to understand why failures happen

When something went wrong (e.g., refusal, empty output, or poor response), it was not always clear why. The key takeaways are:

- Black-box testing alone is not enough to explain root causes
- Having visibility into intermediate steps (model, application logic, guardrails) would make debugging much clearer
- Adding a second-level human-reviewed evaluation layer helps improve accuracy for unclear or borderline cases



Guardrail effectiveness is best assessed at the application level, not just at the model or prompt level

In real-world use, what matters is whether the final user-facing output remains safe, accurate, and useful after passing through the full system. End-to-end testing with realistic inputs helps compare guardrail setups under the same conditions, showing not only how well risks are blocked, but also whether safeguards unintentionally disrupt performance through refusals or empty outputs.

AI VERIFY FOUNDATION