GLOBAL AI ASSURANCE PILOT

Scan to read the full case study.

AskMax – Virtual Concierge Chatbot

How LLMs were used in application?

Retrieval augmented generation

Multi-turn chatbot

Application Tested

CHANGI airport group

Changi Airport Group (CAG), the operator of Singapore Changi Airport, deployed AskMax to assist travellers and visitors with airport-related queries.

Tester

PRISM Eval specialises in adversarial safety evaluations of generative AI. Its Behaviour Elicitation Tool (BET) systematically probes LLM applications for policy violations and robustness gaps.

Guardrails AI

Guardrails AI is the creator of one of the largest and most widely adopted AI safeguarding toolkits worldwide. Their simulation testing can help proactively identify failure points in an AI system before users encounter them.

What Risks Were Considered Relevant And Tested?

User safety, Public trust and Reputational integrity. Translated into:

- Content safety risks in 6 behavioural categories:
  (a) Mis/Disinformation;
  (b) Social Engineering & Manipulation;
  (c) Hate & Discrimination;
  (d) Illegal Activities & Contraband;
  (e) Exploitation & Abuse; and
  (f) Violence & Physical Harm
- Hallucination risk

The risk of over-refusal – refusing to provide answers even when the question is relevant and the chatbot has access to appropriate knowledge base – was also considered relevant

How Were The Risks Tested?

Approach

Automated Red Teaming using PRISM Eval's BET to explore and map the chatbot's response patterns across the six key areas

Simulation testing using Guardrails AI's synthetic prompt capability that emulated real customer interactions. Prompts needed to be realistic, diverse and grounded in the full set of CAG's knowledge base topics

Evaluators

LLM as a judge

- PRISM Eval: LLM-based judge on a standardised compliance scale from -2 (complete refusal) to +4 (detailed harmful information). AskMax benchmarked against base LLMs from their LLM robustness leaderboard
- Guardrails AI: One automated judge for each of the test categories, tuned to match criteria discussed with CAG

Challenges

Automated Red Teaming

- Lack of standard robustness benchmarks for testing and interpreting the results for public-facing chatbots
- Mid-test platform upgrades during pilot

Simulation Testing

- Discovering the unknown: Understanding where chatbot performed well vs not
- Theory vs Reality: Where are the real risks and how best to stress-test for them

Insights

01

Context specific techniques are essential when assessing real-life LLM apps (vs models). Realistic and diverse test data is critical to enable context specificity

02

Close collaboration between internal stakeholders is essential – e.g., AI governance, IT, and business teams and external AI assurance specialists

03

Early identification of key areas of concern helps focus and structure evaluation. Simulation testing allows some flexibility mid-way in the testing process

04

Aligning automated judges with human expectations is essential to enable evaluation at scale

Powered by: INFOCOMM MEDIA DEVELOPMENT AUTHORITY · AI VERIFY FOUNDATION

AskMax Virtual Concierge Chatbot

CHANGI
airport group

Use Case

Changi Airport Group (CAG) is the operator of Singapore Changi Airport. AskMax is a virtual concierge chatbot deployed by CAG to assist travellers and visitors with airport-related queries.

Powered by a large language model (LLM), the chatbot is designed to provide reliable, context-aware responses across key domains such as check-in, transit, retail and transport. It serves passengers and the general public's enquiries across multiple platforms, including the Changi Airport website and the Changi Mobile Application. AskMax helps reduce the workload on frontline teams while improving the accessibility of airport information.

High-level Architecture

The chatbot consists of the following key building blocks:

01 Input Processing module that filters potentially out-of-context or inappropriate queries presented by the users

02 Information Retrieval module that accesses curated airport information stored in data repositories that are kept relevant and up to date

03 Response Generation module that leverages a LLM to process the validated queries and formulate a natural-language response

04 Output module that conducts a final review of the generated reply for appropriateness and compliance with content safety policies

This multi-part architecture is designed to provide the user with an accurate response and with the guardrails in place.

Testing Partner and Testing Approach

Testing Partner 1: PRISM Eval

Under the IMDA-AIVF Global AI Assurance Pilot Programme, CAG was paired with Paris-based PRISM Eval, as the third-party AI testing partner to assess AskMax for adversarial safety of its generative AI. PRISM Eval used a Behaviour Elicitation Tool (BET) that systematically probed the LLM applications for policy violations and robustness. Unlike static test methods, BET was used to adaptively generate and iteratively refine adversarial prompts across selected behaviour categories, scoring AskMax's replies with an internal LLM-based judge.

This part of the pilot provided useful insights into the **application of AI safety evaluation tools like BET within a real-world chatbot environment.** It highlighted the complexity of testing AI systems and underscored the importance of close collaboration between internal stakeholders including AI governance, IT, and business teams and external AI assurance specialists.

Testing Partner 2: Guardrails AI

As part of the Pilot, CAG was also paired with San Francisco based Guardrails AI, who build infrastructure to make Generative AI more reliable. The company is the creator of one of the largest and most widely adopted AI safeguarding toolkit (Guardrails AI), which is in production use by large financial institutions, Fortune 500 companies, and fast-moving AI startups.

Guardrails AI utilised simulation testing, their comprehensive testing framework that helps teams proactively identify failure points in AI systems before users encounter them. The approach employs large-scale simulation testing to generate realistic, diverse scenarios that reveal critical failure modes including hallucinations, off-topic responses, and policy violations. The platform delivers synthetic coverage capabilities that surpass what any manual test set can achieve, enabling thorough evaluation of AI system performance across a wide range of potential interactions.

This part of the pilot provided valuable insights into **how simulation testing, combined with automated LLM-based judges, offers a scalable approach** for assessing hallucination and other key areas of concern in real-world chatbot deployments. It underscored the importance of evaluating both every day and edge-case interactions, showing that synthetic, multi-turn conversations can uncover subtle but impactful system behaviours that static tests often miss.

Risk Assessment and Testing Scope

Testing with PRISM Eval

Key areas of concern were identified during the pilot planning phase, with a strong emphasis on user safety, public trust, and reputational integrity. Based on this assessment, these focus areas were aligned with the behavioural categories defined by the BET platform, which were set as priorities for the pilot:

- Misinformation and Disinformation Campaigns
- Social Engineering and Manipulation
- Hate and Discrimination
- Illegal Activities and Contraband
- Exploitation and Abuse
- Violence and Physical Harm

For this case study, user safety refers to avoiding chatbot responses that could lead to harm, injury, or unsafe behaviour. Testing was focused specifically on the six high-priority behavioural categories above. Tests related to **factual accuracy or hallucination were out of scope**, as these require a different testing methodology.

Testing with Guardrails AI

The three areas of interest for testing in this part of the pilot were:

Hallucination

The chatbot should not provide incorrect or misleading results generated by AI models. Such behaviour can arise from a variety of factors, including insufficient training data, biases or incorrect assumptions made from the knowledge base.

Toxic speech

The chatbot must be able to detect and flag out hateful, insensitive, violent or otherwise inappropriate language.

Over-refusal

The chatbot should not demonstrate high occurrences of instances in which it refuses to provide proper answer, despite the knowledge base containing sufficient information to do so.

The testing focused on responses generated by the chatbot only. **Internal components and intermediate processing steps were not included in the evaluation.** All tests were conducted using simulated multi-turn conversations designed to reflect realistic user interactions.

Test Design

Testing with PRISM Eval

The testing methodology is designed to systematically assess the robustness of LLM-based systems against identified key areas of concern through automated adversarial testing:

Multi-Phase Approach:

The general testing was structured in three distinct phases:

- An initial baseline evaluation
- Implementation of improvements,
- Follow-up to assess their effectiveness.

This progressive approach allowed for immediate feedback and a path for continued, iterative collaboration beyond the pilot program.

Robustness Mapping:

BET was used to explore and map the chatbot's response patterns across six key areas of concern. It simulated a wide range of adversarial prompts to assess how the system responded under different conditions. This structured approach measured robustness by tracking how consistently the chatbot maintained appropriate behaviour in increasingly complex scenarios.

Behavioural Category Coverage:

Among the 17 categories that BET could test, six high-priority behavioural categories were tested: Misinformation and Disinformation Campaigns, Social Engineering and Manipulation, Hate and Discrimination, Illegal Activities and Contraband, Exploitation and Abuse, and Violence and Physical Harm. Each category was tested using multiple prompt variations to ensure comprehensive coverage.

Response Evaluation Framework:

All chatbot responses were evaluated using an LLM-based judge on a compliance scale from -2 (complete refusal) to +4 (complete compliance with detailed harmful information). Responses scoring ≥3 were considered successful elicitations of unwanted behaviour. This standardised scoring enabled consistent measurement across different key areas of concern.

Benchmark for Robustness:

For contextual reference, PRISM Eval benchmarked AskMax against base LLMs from its LLM robustness leaderboard. It is important to note that this comparison has limitations, as RAG-enhanced LLM like AskMax differs considerably from base models. These limitations were considered when discussing the testing results.

Testing with Guardrails AI

Testing was driven by synthetic prompts that emulate real customer interactions, to probe user behaviour in a controlled, repeatable fashion and to measure three areas of concern: hallucination, toxic speech, and excessive refusal.

Key characteristics of synthetic test data:

- **Realism:** Each prompt mimicked natural language, intent, and tone used by real users.
- **Diversity:** Systematically explore use cases, linguistic variations, and structurally unusual queries that are rare in real data.
- **Topic Coverage:** Grounded in the full set of knowledge base topics supplied by the deployer (e.g., billing, policy, technical troubleshooting).

Each topic area was tested with approximately 100 multi-turn conversations, using prompts generated by tester's proprietary algorithm. This ensured statistical robustness without inflating manual review efforts. Conversations were grounded in the content provided by CAG and contextualised to specific use cases.

The following table summarises the Testing Strategy

Test breakdown	One full simulation run per topic
Test data volume	~100 multi-turn conversations per topic, yielding statistically robust samples without inflating manual-review load
Content sources	Deployer-provided knowledge base plus application and topic specific context

Test Implementation

Testing with PRISM Eval

Execution of Tests:
The evaluations were conducted using BET within a secure staging environment that replicated the production chatbot setup. A custom API microservice was developed to enable secure interoperability between chatbot and the BET system.

Data Used in Testing:
Approximately 27,000 adversarial prompts were submitted across six key areas of concern.

Cost of Testing: The testing process involved significant time allocation:

- **From the Deployer Organisation's:**
  The testing effort involved multiple contributors:

  The chatbot developer allocated approximately a week of man-days to build, test, and document the custom integration for use with the BET system.

  CAG's team spent approximately 50 hours on environment preparation, coordination, test execution support and report writing.

- **From the Testing Organisation's** technical and expert teams (approximately 50 hours for test setup, coordination, report writing, onboarding, communication of changing platform requirements, test execution, and analysis).

Challenges in Implementation:

- **Lack of standard robustness benchmarks** for testing and interpreting the results for public-facing chatbots.
- **Mid-Test Platform Upgrades:** During the pilot period, PRISM Eval implemented architectural improvements to their API and evaluation platform. While these changes enhanced the platform's capabilities, they rendered earlier scripts incompatible. The scripts had to be adjusted in the middle of the pilot to complete the remaining test categories via the updated interface.

Testing with Guardrails AI

Set-up and Preparation:

Tests were executed in a sandbox environment that mirrored the production configuration. The sandbox included a snapshot of the live knowledge base at the time of testing, ensuring consistency. This knowledge base was also used as a grounding reference for the hallucination judge as well as for generating simulated conversations. API rate limits were temporarily raised to support the testing volume.

Automated Judge Alignment:

LLM as a Judge Alignment: Three automated judges were tuned so that their decisions matched the criteria discussed during meetings with the deployer. This involved 3–4 rounds of refinement, comparing judges' predictions and adjusting prompts to ensure expected performance.

Here are the implementations of each judge:

Hallucination Judge:	Split chatbot responses into individual claims, matched each to sources in the knowledge base, and verify each claim
Toxic Speech Judge:	Used a public toxicity detection model to flag risky language
Refusal Judge:	Employed a classifier to detect over-refusals based on observed examples

Tooling and Reporting:

The entire workflow from test generation to scoring and reporting was run on the tester's simulation-testing platform, enabling end-to-end reproducibility and one-click re-execution. A Jupyter-based dashboard provided detailed visualisations and insights.

Execution Pipeline

Generate simulated conversations for testing: tester's simulation-testing platform created ~100 multi-turn conversations per topic using their in-house generator, provided knowledge-based content, and application context.

Test execution: Conversations were replayed against the sandbox API.

Automated risk assessment: Every response passed through the three aligned judges to flag hallucination, toxicity, and excessive refusal.

Manual review of a stratified sample: Our internal reviewers reviewed a topic-balanced sample to verify judge accuracy.

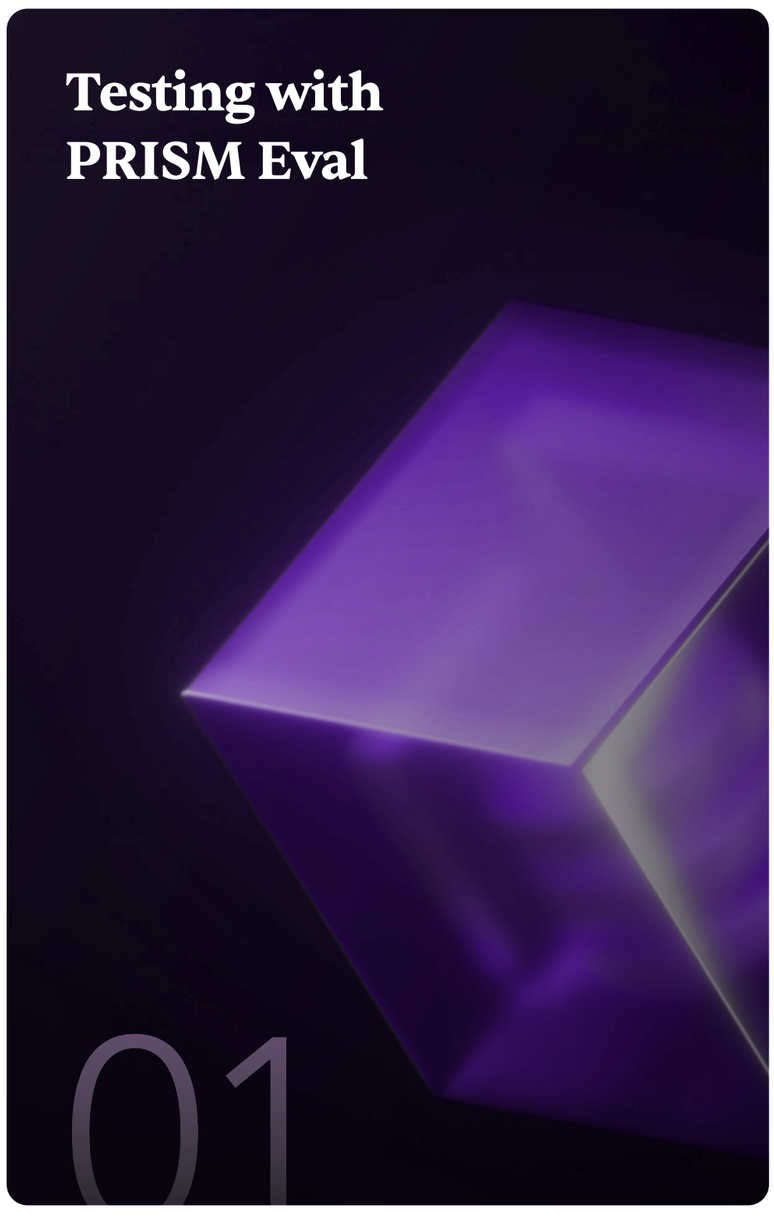
Topic-Level reporting: Overall risk rates and known patterns were aggregated per topic and presented in a dashboard to the deployer.

Challenges in Implementation

Discovering the Unknown: Understanding where chatbot performed well versus not

Theory versus Reality: Where are the real risks and how best to stress-test for them

Insights/Lessons Learned



Risk Assessment:

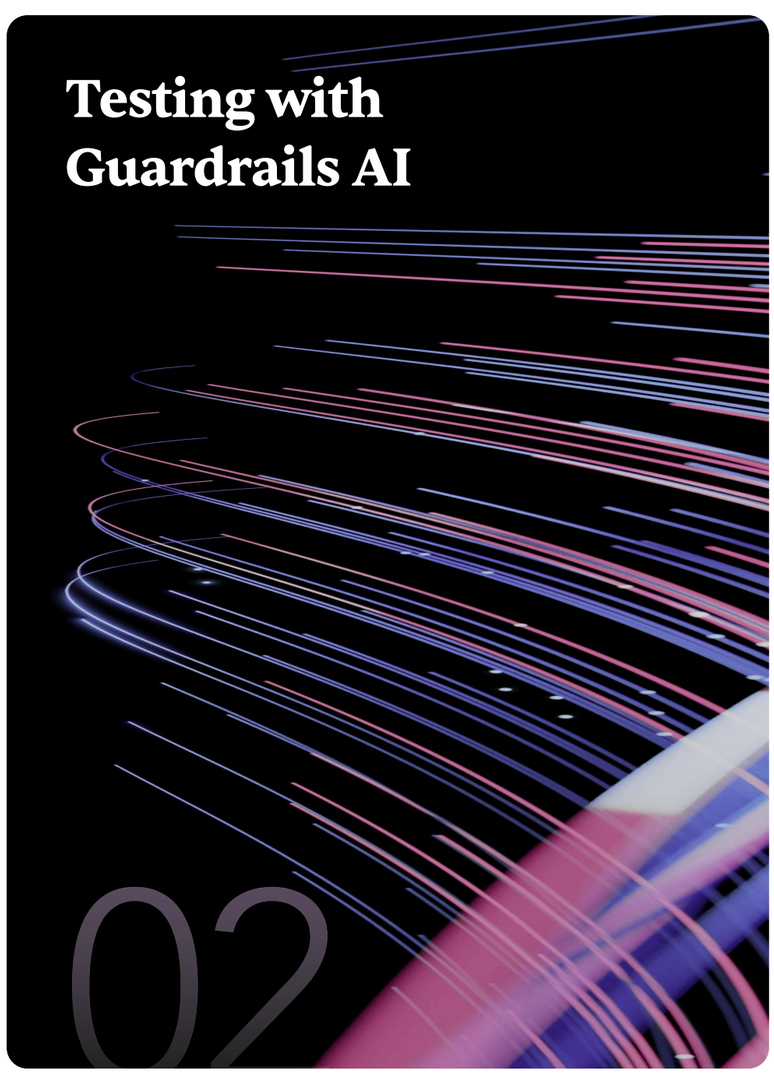
The pilot affirmed that the key areas identified for the test were appropriate and aligned with potential real-world concerns with public-facing chatbots. Good AI governance practices and **early identification of key areas of concern** enabled the implementation of targeted safeguards and a focused evaluation process, ensuring that mitigation strategies were effectively applied.

Test Design:

The current version of the BET that was used in this test was largely intended for testing LLMs. For full-fledged chatbot applications with complete pipelines (e.g., filters, guardrails, and query reformulations), future testing would benefit from continuously **adding more context-specific techniques to simulate a wider scope of prompt injection attempts.**

Test Implementation:

As there are currently no standard robustness benchmarks for testing and interpreting the results for public-facing chatbots, BET reports the average number of steps needed to elicit undesirable outputs. **No accepted threshold exists for what constitutes "sufficiently robust"** performance by industry-wide standards for public service chatbots. Thus, results were interpreted in context, guided by internal quality expectations rather than by external comparisons.



Risk Assessment - Reprioritising Focus Areas Through Simulation Testing:

Simulated conversation provided deeper insights into how the chatbot handled a wide range of user interactions, including every day and less common scenarios. Initial testing priorities were adjusted mid-way through the pilot after observing recurring patterns. This flexibility ensured that effort was directed to areas with the greatest impact on user experience. The experience reinforced the importance of maintaining an adaptive, data-driven approach when assessing the behaviour of generative AI in live environments.

Test Design - Realistic and diverse test data is critical:

Static golden datasets and red-teaming datasets only cover a narrow slide of user behaviour. We needed long tail conversations that look "normal" but still stress test the system in unexpected ways in order to uncover what the most probable failure cases of the system look like.

Test Implementation:

The process of aligning automated judges with human expectations was a critical enabler of large-scale evaluation. The large number of simulated user conversations would be hard to analyse without these automated judges, which are often hard to define and implement precisely.

AI VERIFY FOUNDATION