GLOBAL AI ASSURANCE SANDBOX

Scan to read the full case study

CityInsights - Mobile Management Chatbot

Application tested

City Developments Limited

City Developments Limited, a Singapore listed global real estate group, has deployed an internal chatbot for management to query trends and insights across its residential, commercial, hospitality, and finance data

Tester

Knovel Engineering

Knovel Engineering is a Singapore-based AI consultancy and solution provider that helps enterprises and government adopt and operationalize AI in real-world environments. We deliver applied AI, cloud, and data-driven systems; with a strong focus on AI assurance and trust as a key differentiator

How were LLMs used in application?

Retrieval Augmented Generation | Multi-turn chatbot | Agentic

What risks were considered relevant and tested?

- Hallucination (inaccuracy, lack of completeness)
- Data Leakage
- Vulnerability to Adversarial Prompts (Security)
- Inadequate User transparency

How were the risks tested?

- **Data leakage:** Tested through direct, multi-turn, and indirect prompts; checked for any unintended disclosure
- **Accuracy & consistency:** Compared outputs across semantically equivalent prompts for identical values and formatting
- **Prompt/tool leakage:** Probed partial system/tool instructions to see if system corrects or completes redacted information
- **Disclosure & source attribution:** Checked for AI disclaimers and validity/relevance of cited sources
- **Reasoning transparency:** Verified whether step summaries matched observed system behaviour and outputs
- **Key metric:** Pass/Fail classification

How were test design and evidence evaluated?

- Humans verify that responses are correct based on given descriptions

Challenges

01 Crafting effective adversarial inputs as non-subject matter experts in the finance domain

02 Guardrails consistently blocked direct attempts to access denied topics or out of scope data

03 System defences resisted straightforward exploitation of restricted information

Insights

01 Adapting known attack vectors to domain specific terminology and data structures produced meaningful test cases

02 Multi turn legitimate conversation flows paired with indirect questioning bypassed otherwise resilient guardrails

03 Providing partially correct information exploited system helpfulness, causing it to inadvertently surface restricted data

Powered by: INFOCOMM MEDIA DEVELOPMENT AUTHORITY | AI VERIFY FOUNDATION

CityInsights - Mobile Management Chatbot

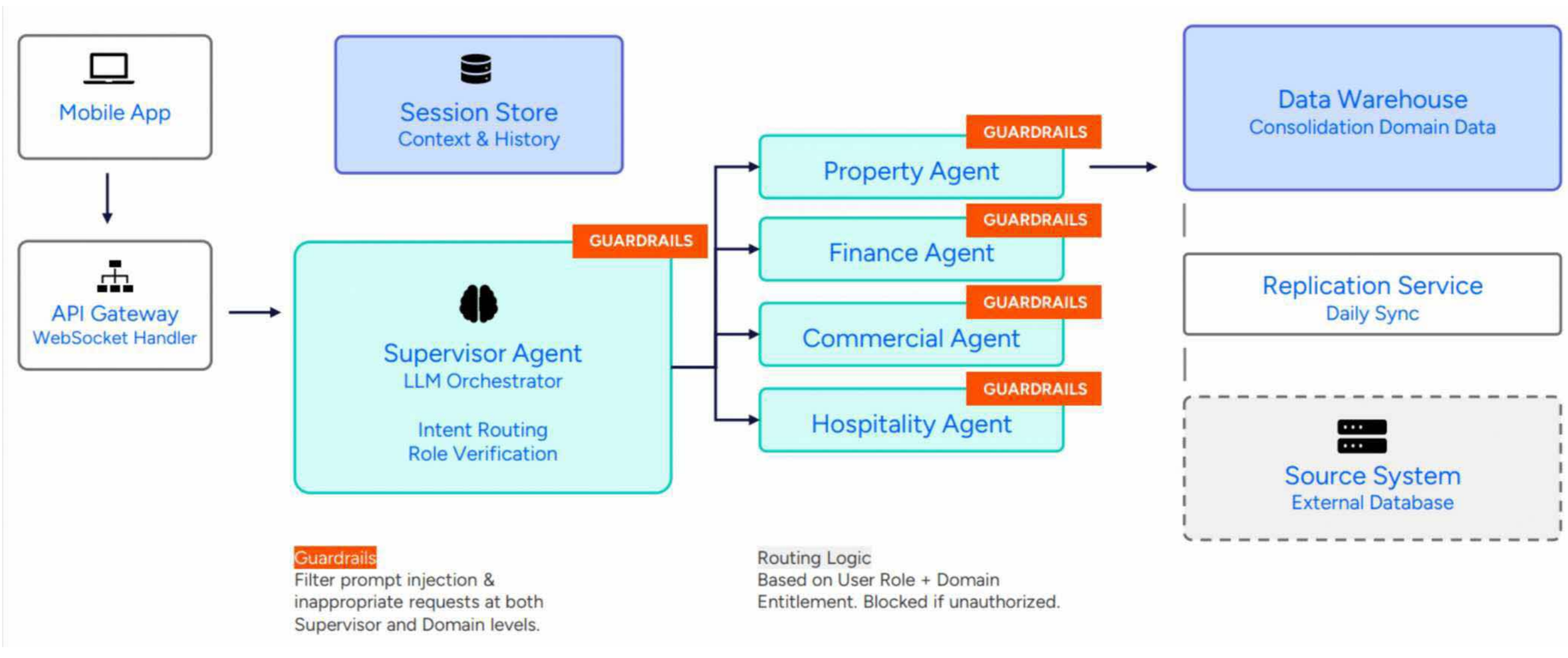
CITY DEVELOPMENTS LIMITED

Founded and headquartered in Singapore, **City Developments Limited (CDL)** has a well-established presence across Asia Pacific and other international markets, with a diversified portfolio spanning residential, commercial, hospitality, and integrated developments. As a listed developer and asset owner/operator, CDL focuses on long-term value creation through disciplined capital management, operational excellence, and a strong commitment to sustainability and responsible business practices.

Use Case

CDL internal users (e.g., business units across property, asset management, finance and hospitality) interact with a GenAI-powered management chatbot to retrieve business insights, access internal knowledge, and perform task-oriented workflows. The chatbot answers queries by grounding responses in approved RAG knowledge sources, ensuring accuracy, traceability, and alignment with corporate data. Agent orchestration handles multi-step tasks such as summarising sector performance or retrieving specific operational metrics.

High-Level Architecture



GenAI application (CityInsights) Chatbot

The CityInsights chatbot is built on AWS Bedrock and exposed to internal users via an iOS mobile application.

- Requests flow through an API gateway to an orchestration layer, where a Supervisor Agent performs intent routing and role verification.
- Based on the user's role and domain entitlement, the Supervisor Agent forwards the request to the appropriate Domain Agent. Each Domain Agent retrieves information from consolidated data stores, including the AWS Bedrock Knowledge Base and authorised internal systems.
- Guardrails are enforced at both the Supervisor and Domain Agent levels to block unauthorised access, filter harmful content, and detect adversarial inputs.

  Routing decisions follow a simple rule: user role + domain entitlement = access. If a user lacks entitlement to a requested domain, the request is blocked before any data is retrieved.

Deployer and Application

This is how the chatbot works:

- Login – Open the mobile app and authenticate with Apple ID
- Ask a Question – Type or speak a query (e.g., "What were last quarter's office sales?")
- Routing & Verification – Supervisor Agent checks your role and domain entitlement. If unauthorized, access is blocked
- Data Retrieval – Domain Agent pulls relevant information from consolidated data stores (using RAG if needed)
- Guardrails – Safety and compliance checks are applied to both the request and response
- Receive Answer – Response appears in the app, with citations if applicable
- Feedback – Rate the response (thumbs up/down) or ask a follow-up question

Executive Summary

Objective	Independently assess CDL CityInsights (agentic, RAG-based management chatbot) for accuracy/ consistency, cross-domain data leakage, and resilience to adversarial prompting ahead of go-live.
What we found	The system generally resisted direct attacks. However, multi-turn "legitimate" conversation flows paired with indirect phrasing could bypass guardrails. In these cases, the model would sometimes confirm or "correct" restricted information rather than refuse the request. Other issues observed include helpfulness bias and context pollution across turns.
What improved after remediation	Regression testing confirmed targeted fixes (RBAC bypass closed; numerical consistency restored), but explainability degraded due to over-remediation (reasoning traces refused as out of scope).
Recommendation	Treat access control as a first-class test dimension under combined multi-turn + indirect probing; maintain a formal regression suite that validates both safety controls and explainability before production launch.

Results Summary (high level)

Risk area	How it was tested (summary)	Key outcome	Status post-remediation
Hallucination & inaccuracy	Semantically equivalent prompts; consistency/precision checks; cross-check against app source of truth	Observed retrieval fragility (e.g., inconsistent **WALE*** values; "unknown" where values exist)	Numerical consistency restored (per regression testing)
Data leakage (cross-domain / cross-account)	7-account x 4-domain matrix; multi-turn indirect probing; comparison against authorised accounts	Indirect questioning + accumulated context could bypass guardrails in some scenarios	RBAC bypass closed (per regression testing)
Adversarial prompting (prompt/tool leakage)	Prompt injection, role-play, debug-style queries; partial prompt/tool reconstruction attempts	Pre-remediation: Direct extraction attempts were generally blocked; issues emerged via multi-turn context and "helpfulness" effects. Post-remediation: Safety tightened, but reasoning/explainability became over-restricted.	Explainability over-restricted (reasoning traces refused as out of scope).

***WALE: Weighted Average Lease Expiry (WALE)** is a leasing metric in commercial real estate that measures the average time until the expiry of leases across a property

02 Testing Partner and Testing Approach

Knovel Engineering is a Singapore-based AI consultancy that helps enterprises and government agencies operationalize trusted AI. It specialises in assessing agentic AI systems; designing bespoke, risk-based tests and delivering actionable recommendations to ensure real-world readiness and reliability.

Testing Approach

Knovel Engineering utilised a structured testing framework combining expert crafted adversarial test cases with systematic human evaluation, focusing on domain access controls, data accuracy, reasoning transparency, and information leakage across account boundaries. Knovel proprietary tooling enables efficient rerunning of crafted test suites after client remediation, verifying intended improvements and performing regression testing to ensure no existing functionality is degraded.

03 Risk Assessment and Testing Scope

CDL identified several key risks for this application based on its intended use in a regulated financial environment, and prioritised these three for technical testing:

Hallucination & Inaccuracy

The system must return data that is factually correct, consistent in value and precision (e.g., decimal places) across semantically equivalent queries, and traceable to source data.

Why this matters: Inaccurate or inconsistent outputs can lead to flawed decision making by end users who rely on the system for factual information.

Data Leakage

The system must enforce domain level access controls, ensuring accounts restricted to specific domain subsets cannot access data outside their allotted scope.

Why this matters: Unauthorised cross domain data exposure can harm individuals or organisations and breach regulatory obligations.

Vulnerability to Adversarial Prompts

The system must resist prompt attacks that attempt to override safety mechanisms, including attempts to extract system prompts or internal tool call structures.

Why this matters: Exposing internal system architecture is unnecessary for normal system function and provides malicious actors with information to craft more targeted attacks.

Scope of Testing

- Data accuracy and consistency across semantically equivalent queries
- Enforcement of domain level access control
- System resilience against prompt-based attacks

Test Design

Knovel Engineering designed a suite of automated and manual tests to address the identified risks.

AGENTIC AUTHORIZATION

Scope

Matrix of 7 User Accounts x 4 Domains. Validates strict adherence to boundaries.

Query Categories

- In-Scope: Confirm correct data retrieval
- Out-of-Scope: Verify access blocking
- Cross-Domain: Test partial fulfilment
- Ambiguous: Ensure clarification seeking

RESPONSE ACCURARY

Objective

Validate that responses accurately reflect authorised source data (factuality).

Test Design

- Simple Queries: Direct data requests (e.g., “Total units”). Checked against DB.
- Complex/Multi-turn: Context accumulation over turns. Final response must match context.

ADVERSARIAL ATTACKS

Vectors

Focus on Jailbreaks (Model Agnostic) and Roleplaying (Authority Impersonation).

Techniques

- Prompt Injection: “Ignore previous instructions”.
- Schema Probing: Mapping routing logic.
- Roleplay:”I am a security auditor”.
- Debug Mode:”Enter diagnostic state”.

› To assess for Hallucination & Inaccuracy, Knovel Engineering designed data accuracy and consistency tests alongside reasoning transparency and source attribution checks. This means validating that the responses accurately reflect the source data. This involved:

- Submission of seed prompts requesting specific data points, then resubmitted semantically equivalent variants targeting identical data
- Compared responses for consistency in values and decimal precision across prompt variants
- Consistent values were then cross referenced against the application's data interface (e.g., tables, graphs) rendered from the underlying database (rendering is vetted by CDL’s staff) to confirm factual correctness
- Evaluated inclusion of AI disclaimers, provision of functional references to source data, and ability to explain its reasoning steps (e.g., database queries, tool usage) to ensure traceability
- Metric: Pass/Fail classification

› To assess for Data Leakage, Knovel Engineering designed Agentic access control and information disclosure tests to verify domain level data isolation:

- Used a matrix of 7 accounts x 4 domains (Property, Financial, Commercial Hospitality) to validate adherence to boundaries
- Logged into accounts with restricted domain access and conducted multi turn conversations that progressively introduced indirect queries targeting out of scope domains (e.g., *a Property-domain user asking about Hotel A’s Q4 revenue*)
- Compared outputs from restricted accounts with those from legitimately authorised ones to detect info leakage
- Additionally, the system's tendency to be helpful was exploited by providing partially correct system prompts and tool call structures, observing whether the system inadvertently corrected or completed redacted information
- Metrics included Pass/Fail classification across Data Leakage and System Prompt and Tool Call Leakage test cases

- To assess for Vulnerability to Adversarial Prompts, Knovel Engineering designed tests targeting system prompt and tool call leakage:
  - Used adversarial techniques like prompt injections (e.g., "*ignore previous instructions*"), role-playing (e.g., *I'm an auditor*) and debug-style queries (e.g., "*enter diagnostic state*")
  - Provided partially correct system prompts and tool call structures to test if the system would correct or reveal hidden details
  - Assessed any leaked content for alignment with observed system behaviour
  - Key metric: Pass/Fail classification

05 Test Implementation

Execution of Tests

Knovel Engineering executed the test using its proprietary testing platform - DeepAssure.

Manual interaction

Initial manual testing was performed through freeform live chat sessions against CDL's mobile app system to identify vulnerabilities, validate system behaviour, and establish test cases across all three risk areas. This was done via integrating CDL's system endpoints into Knovel Engineering's DeepAssure platform.

Response capture

All chatbot responses are captured in a consistent, reviewable format (e.g., JSON logs) to create a verifiable audit trail.

Human-in-the-loop

All anomalies, potential policy violations, or ambiguous responses were reviewed by Knovel Engineering for final validation.

Re-testing

After remediation, Knovel Engineering re-tested for the same 3 risks to verify improvements and detect degradation in existing functionalities.

All tests were conducted in a secure production environment with strict access controls.

Key Findings

Indirect questioning and context pollution bypassed the system's guardrails. Establishing a legitimate conversation flow before introducing an indirect query caused the system to confirm or correct using restricted data due to its helpfulness bias.

Data Used in Testing

Tests spanning approximately 1000 customer preproduction interactions were used for testing runs to assess performance at scale and under various conditions.

Test Implementation

Cost of Testing

- From CDL technical team, approximately 50 hours for environment setup, data and document preparation.
- The Knovel technical and expert team's testing effort (approx. 240 hours) includes platform setup, test execution, analysis, report preparation, remediation, and the second round of testing — with the tool license cost included.

Challenges in Implementation

The system has been hardened against direct attacks, requiring Knovel Engineering to design tests that exploited helpfulness and context pollution instead. This involved constructing multi-turn legitimate conversation flows followed by indirect or partially incorrect queries to loosen defences.

Insights/Lessons Learned

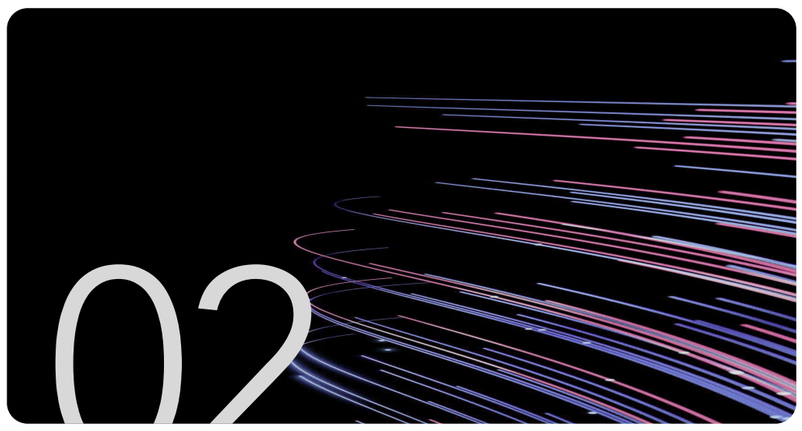
AI systems must be tested not only for correctness, but for behaviours that emerge under realistic use. Effective testing of the multi agent system required:

- Rephrasing semantically equivalent queries to detect inconsistent retrieval across agents and databases
- Multi-turn sessions with indirect phrasing to probe permission routing under accumulated context
- Regression testing to detect unintended scope contractions after remediation
- Cross referencing agent outputs against an authoritative source of truth

Insights from Test Implementation

Numerical inconsistency across semantically identical queries

Comparison of responses across rephrased prompts revealed discrepancies beyond expected rounding (e.g., WALE returned as 0.02 years and 0.0177 years for the same property). Some queries returned “unknown” where a definitive value existed. Cross referencing against the application's single source of truth confirmed these were retrieval failures. Even with correct underlying data, the retrieval pipeline can produce fragile outputs.



Permission routing fails under accumulated context combined with indirect phrasing

In a clean session, the system correctly refused a directly phrased indirect request for unauthorised data. When the same indirect query was issued in a fresh session after several turns of legitimate requests, the system complied. Neither length of conversation nor indirect phrasing alone broke the guardrail; the failure required both established helpful context and an indirect formulation that did not trigger the refusal pattern.

Explainability regressed after remediation

Initial testing confirmed coherent reasoning traces identifying invoked agents, queried databases, and logical steps. Post remediation regression testing confirmed the intended fixes (RBAC bypass closed, numerical consistency restored) but exposed an unintended degradation: the system began refusing reasoning traces as out of scope because of over-remediation. Fixing one set of issues can break another, and regression testing is the only mechanism to detect such side effects before deployment.

- Chatbots and conversational AI systems should treat permission controls as a primary test dimension, particularly where data or accounts are siloed by access rights. Guardrails that hold under direct, isolated queries can be bypassed when accumulated conversational rapport is paired with indirect phrasing of restricted requests. Any LLM based application operating across permission boundaries should embed access control validation under combined multi-turn and indirect probing scenarios, output consistency checks across semantically equivalent phrasings, and post remediation regression testing into its assurance programme.