20 LLM Guardrails

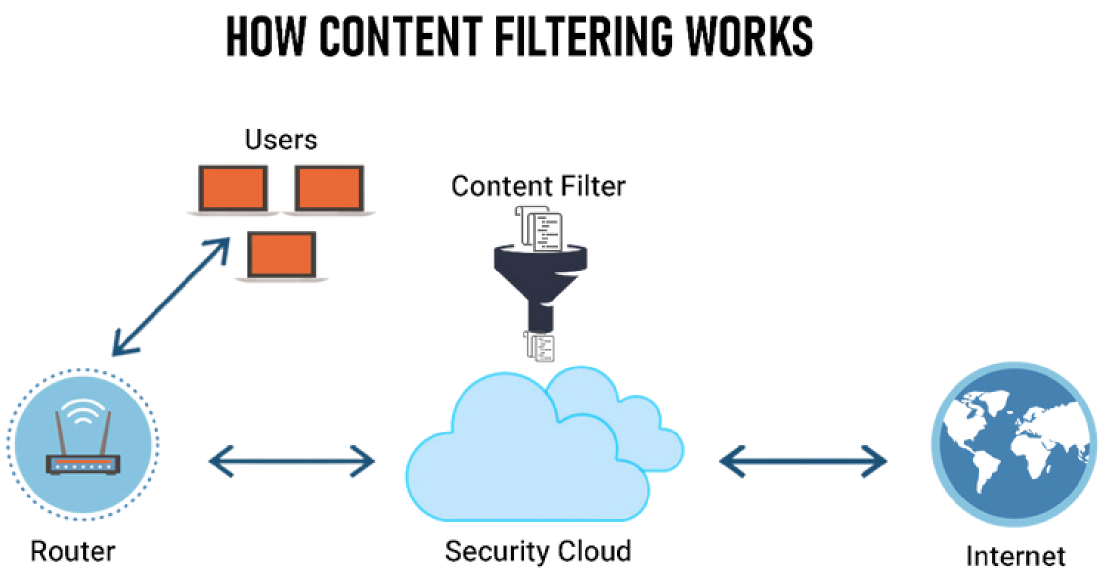
Learn about the 20 essential LLM guardrails that ensure the safe, ethical, and responsible use of AI language models.

→

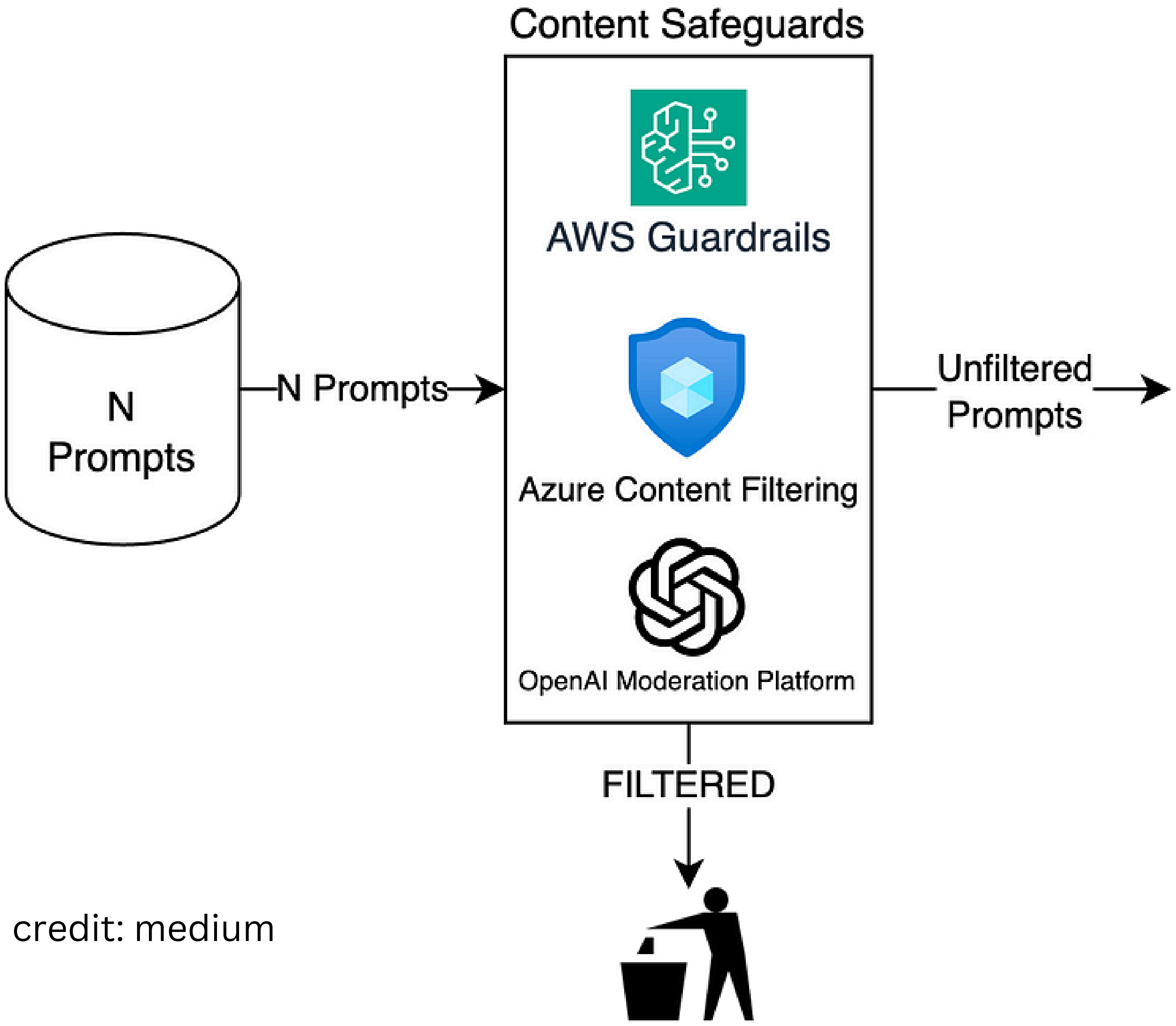
Security and Privacy Guardrails

1. Inappropriate content filter

- Scans for Inappropriate Content: Checks LLM responses for unsuitable words or topics (like NSFW material).
- Uses Smart Models: Combines banned word lists with machine learning to understand context better.
- Blocks or Cleans Output: Flags bad content, either removing it or making it safe before users see it.
- Keeps Interactions Professional: Ensures all conversations stay respectful and appropriate



credit Spiceworks



credit: medium

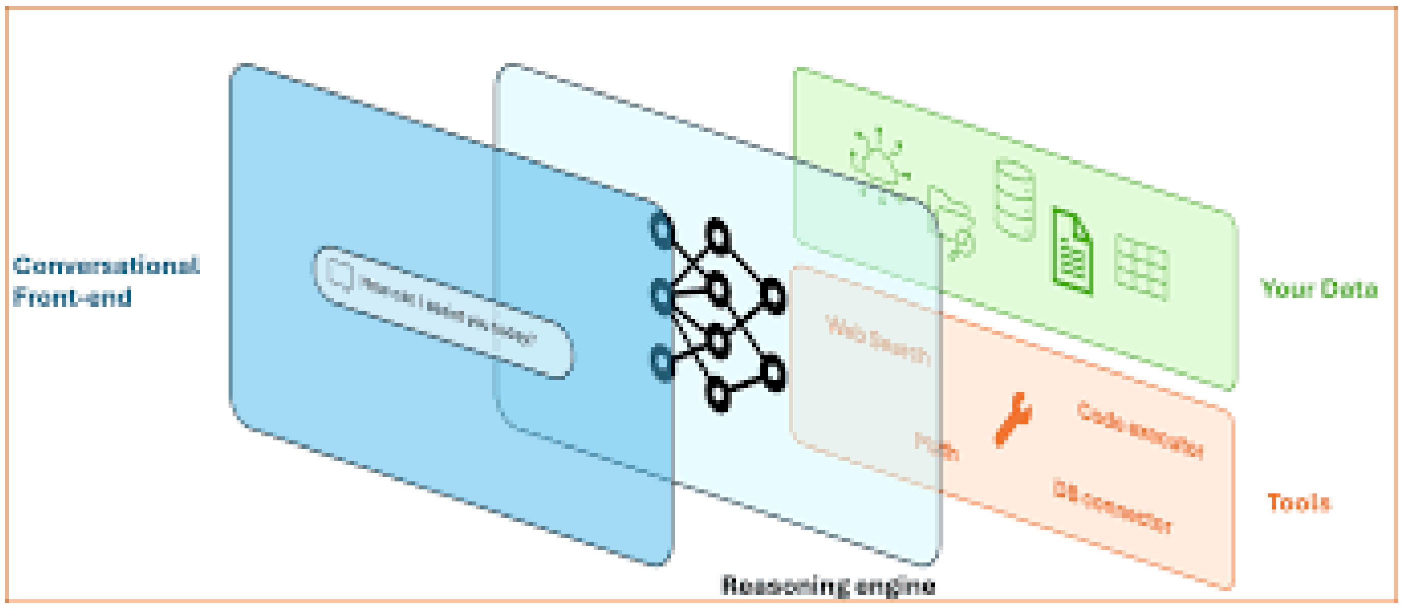
2. Offensive language filter

- Detects Bad Words: Uses keyword matching and smart language tools to spot offensive language.
- Blocks or Edits Responses: Stops or changes flagged content to remove inappropriate parts.
- Ensures Respectful Output: Keeps all replies clean and inclusive, especially for customers.
- Maintains Professionalism: Avoids harmful or rude language in any conversation.

Security and Privacy Guardrails

3. Prompt injection shield

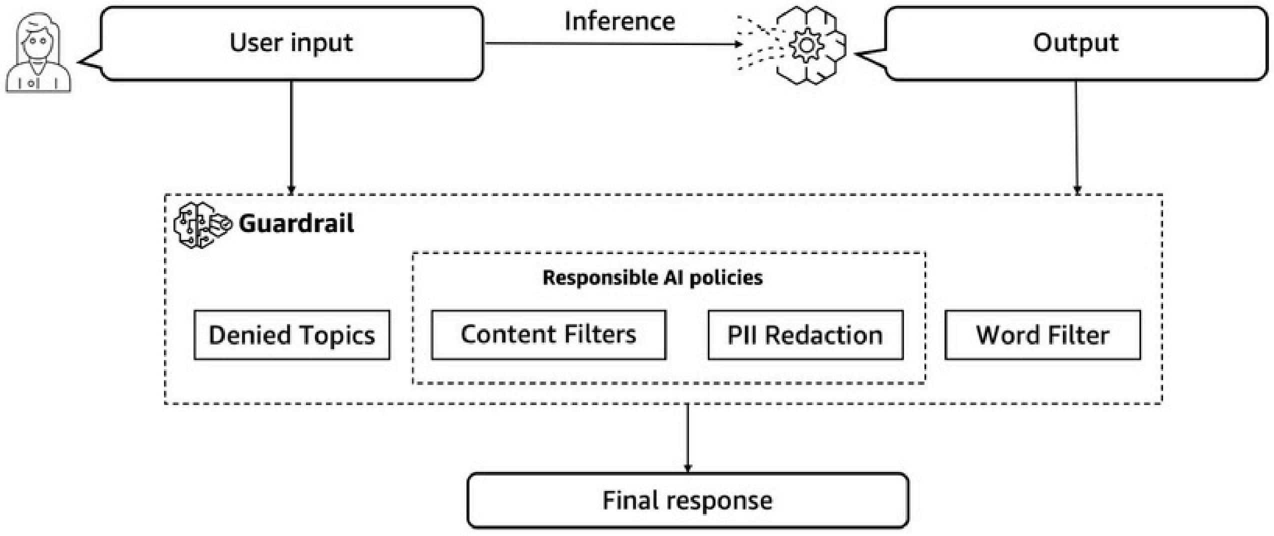
- Spots Sneaky Prompts: Detects tricks to manipulate the model’s behavior.
- Blocks Harmful Requests: Stops inputs that try to make the LLM generate bad outputs.
- Protects System Integrity: Ensures the model follows its rules and stays reliable.
- Keeps Interactions Safe: Prevents misuse by identifying and stopping malicious attempts.



credit: medium

4. Sensitive content scanner

- Detects Sensitive Topics: Uses smart tools to spot controversial or delicate terms.
- Flags or Blocks Content: Stops responses that could be biased or inflammatory.
- Promotes Fairness: Reduces the risk of spreading stereotypes or harmful views.
- Ensures Safe Outputs: Keeps AI responses neutral and respectful on tricky issues.

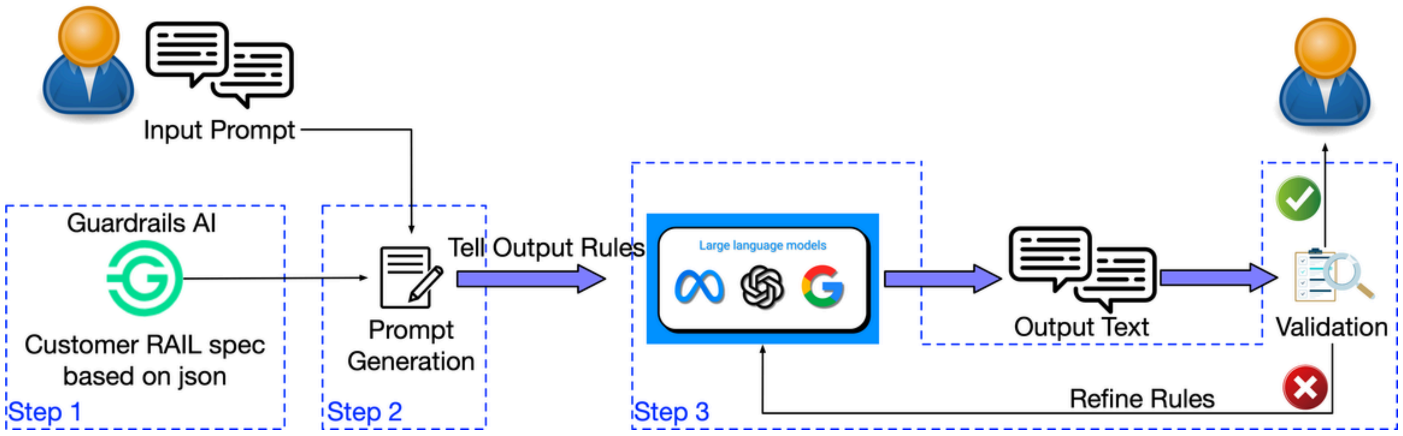


Credit: AWS

Response and Relevance Guardrails

5. Relevance validator

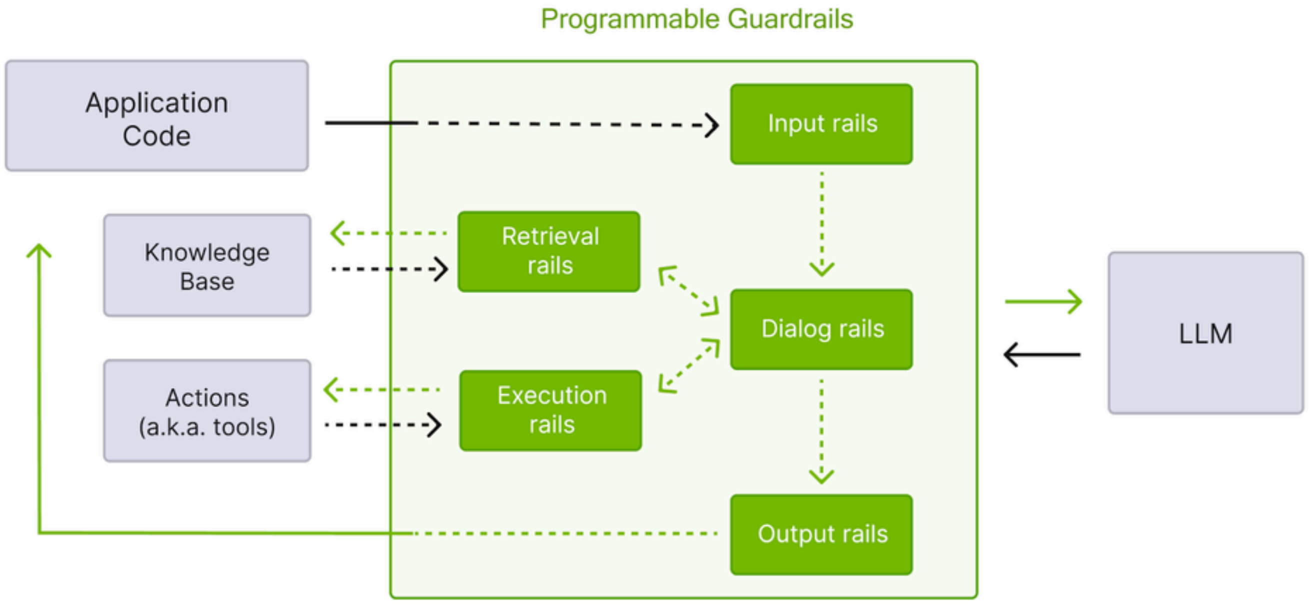
- Checks Topic Match: Compares user input with the response to ensure they align.
- Uses Smart Tools: Leverages advanced models to verify coherence and relevance.
- Fixes Irrelevant Replies: Adjusts or blocks responses that don’t match the question.
- Keeps Answers On-Point: Ensures all replies stay clear and focused on the topic.



Credit: arxiv

6. Prompt address confirmation

- Understands User Intent: Checks if the response aligns with the main idea of the question.
Compares Key Concepts: Ensures the output covers the core points of the prompt.
Improves Completeness: Fills in missing details to provide thorough answers.
Prevents Topic Drift: Keeps replies focused and relevant to the user’s query.



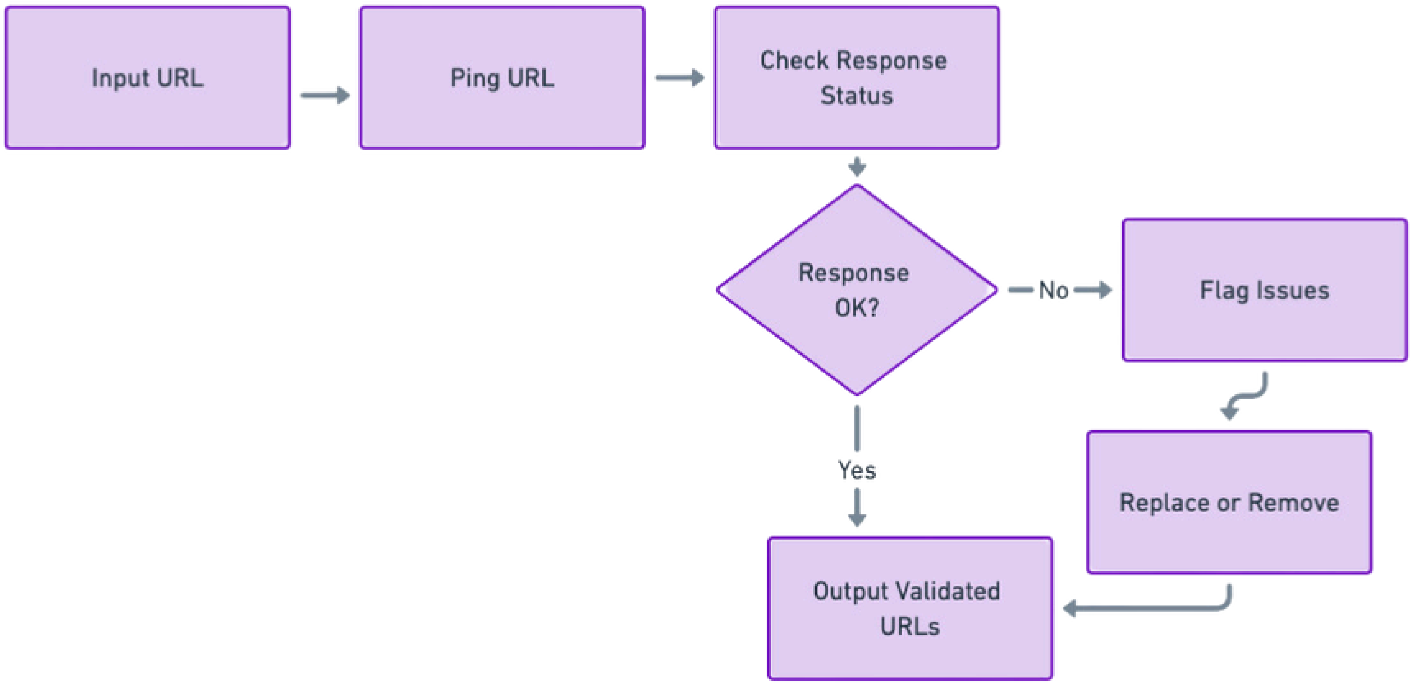
High-level flow through programmable guardrails.

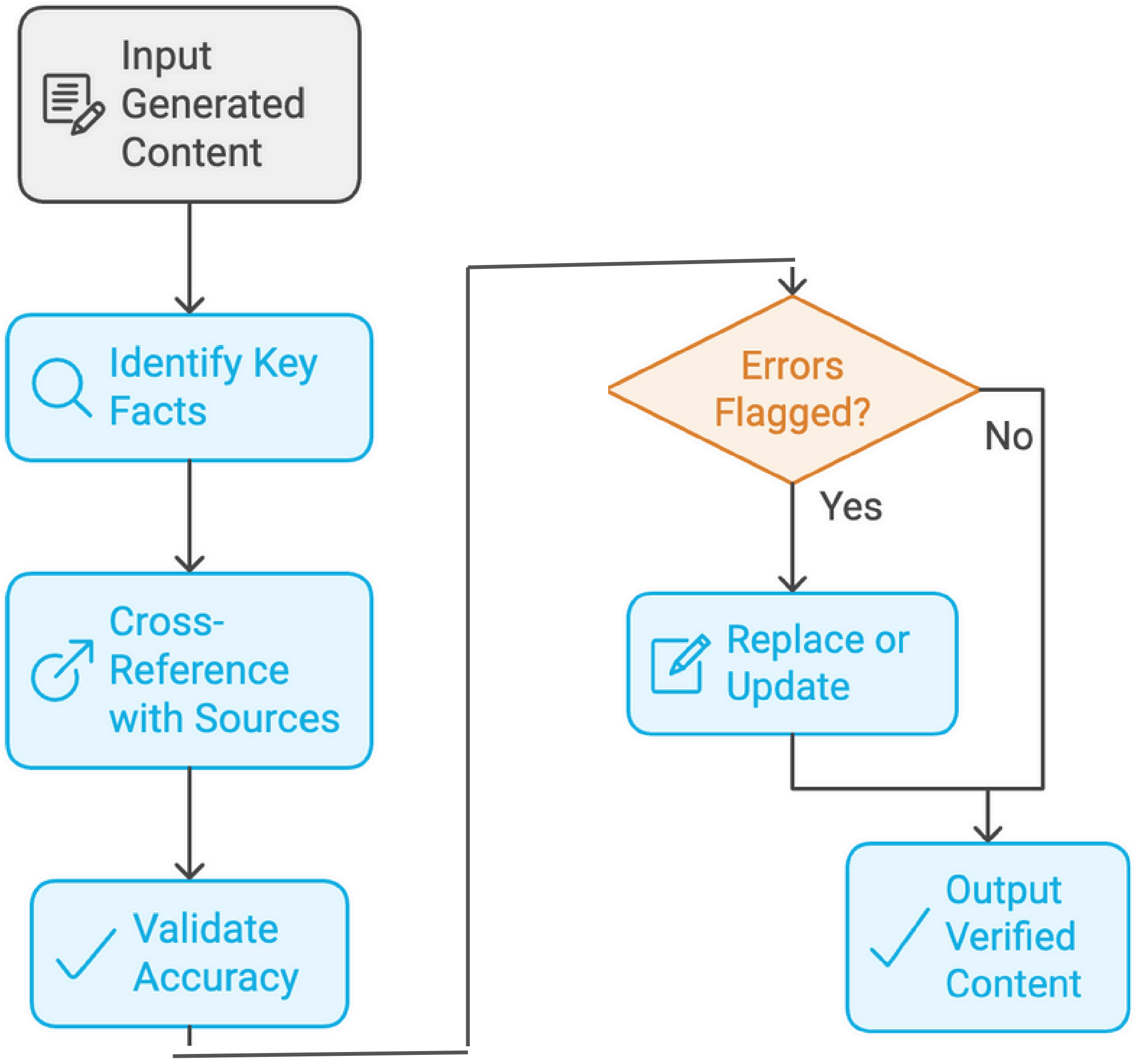
Credit: NVIDIA

Response and Relevance Guardrails

7. URL availability validator

- Checks Link Validity: Verifies if suggested URLs are live and working.
- Uses Real-Time Status: Pings web addresses to confirm their status.
- Removes Broken Links: Flags and excludes invalid or unsafe URLs.
- Keeps Responses Reliable: Ensures users get accurate and safe links.





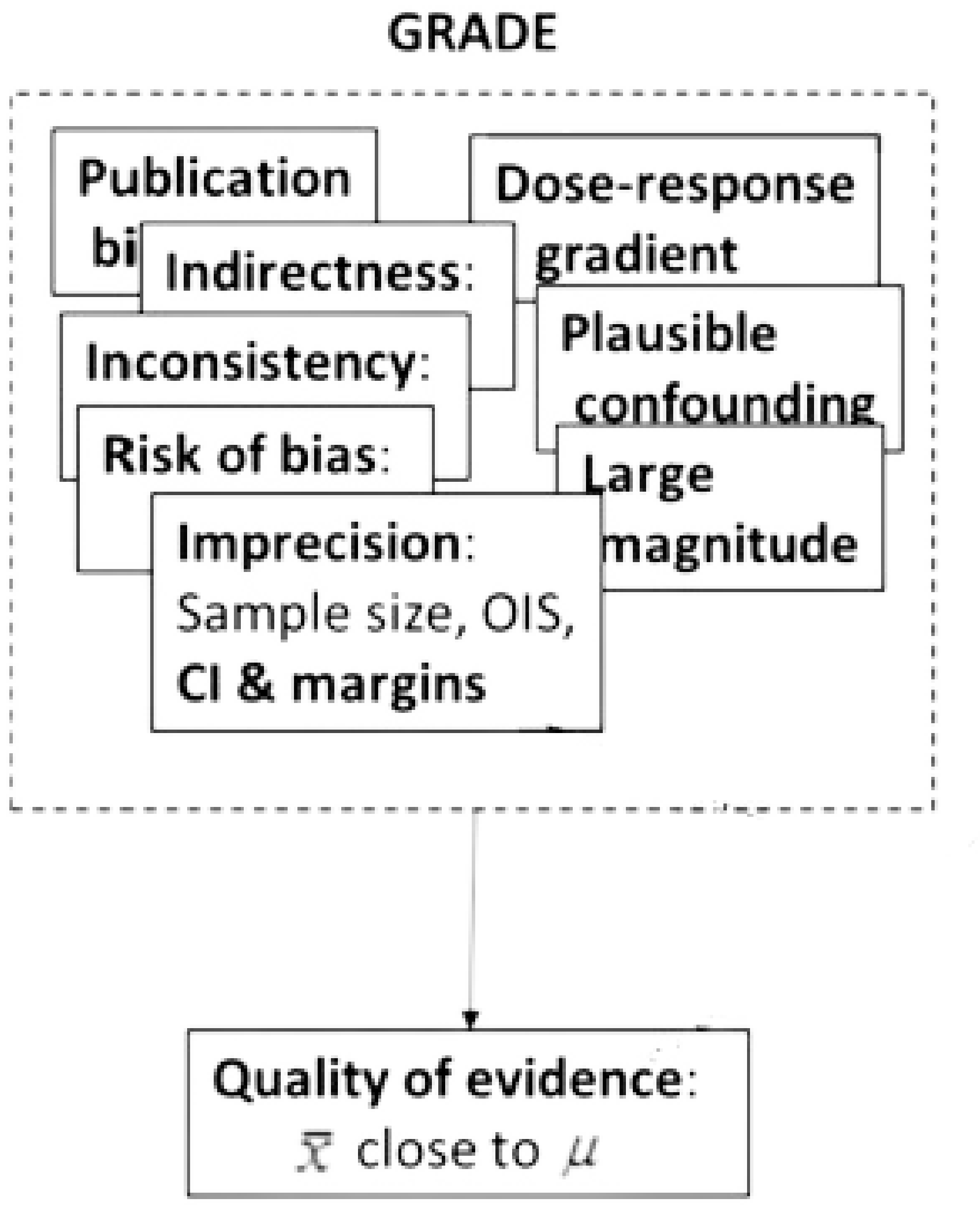
8. Fact-check validator

- Verifies Accuracy: Cross-checks generated facts with trusted sources.
- Uses External APIs: Leverages up-to-date knowledge for validation.
- Corrects Misinformation: Replaces outdated or wrong facts with verified data.
- Builds Trust: Ensures LLM responses are factual and reliable

Language Quality Guardrails

9. Response quality grader

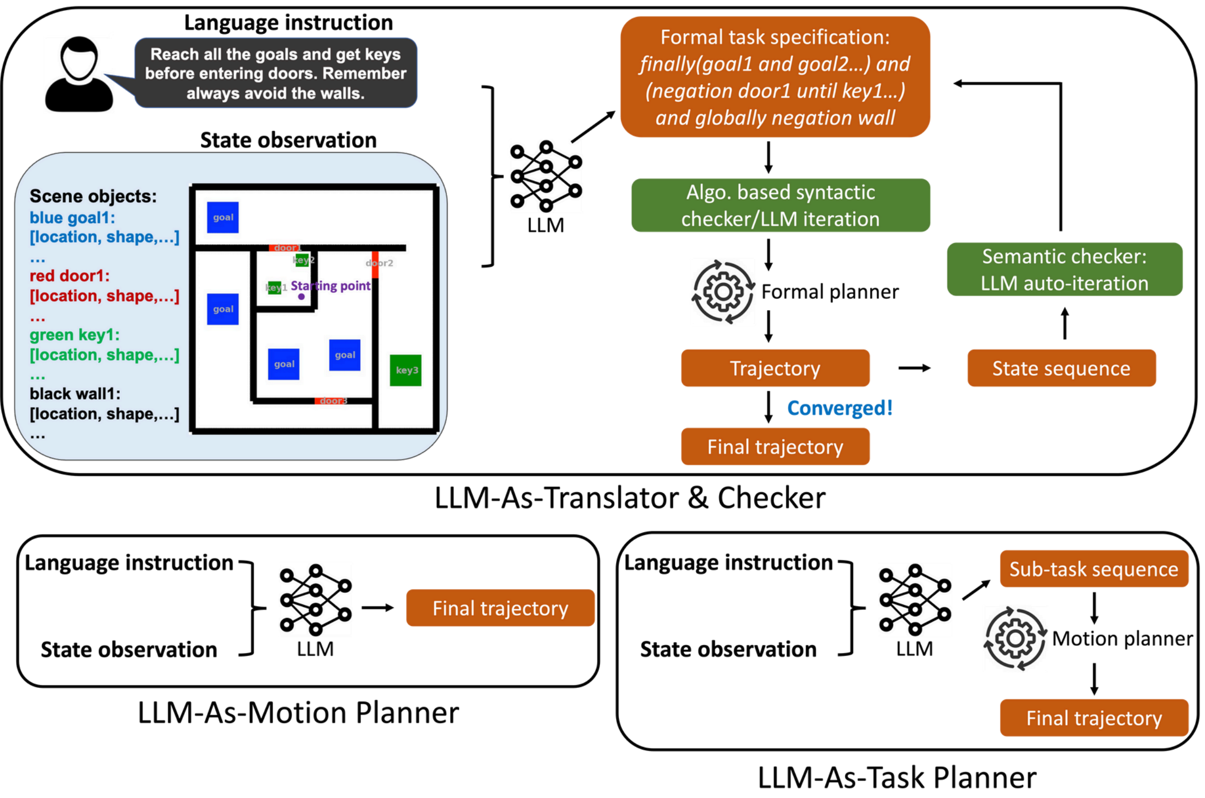
- Checks Output Quality: Reviews if the response is clear, relevant, and well-structured.
- Uses Smart Models: Scores responses based on examples of good writing.
- Flags Poor Replies: Identifies unclear or messy answers for improvement.
- Ensures Readability: Suggests changes to make replies easy to understand



Credit: ScienceDirect.com

10.Translation accuracy checker

- Verifies Translations: Ensures the translated text is accurate and meaningful.
- Checks Context: Confirms the translation preserves the original intent.
- Uses Language Databases: Cross-references translations with trusted sources.
- Fixes Mistakes: Corrects any errors to ensure accurate multilingual communication.



LLM-As-Translator & Checker

LLM-As-Motion Planner

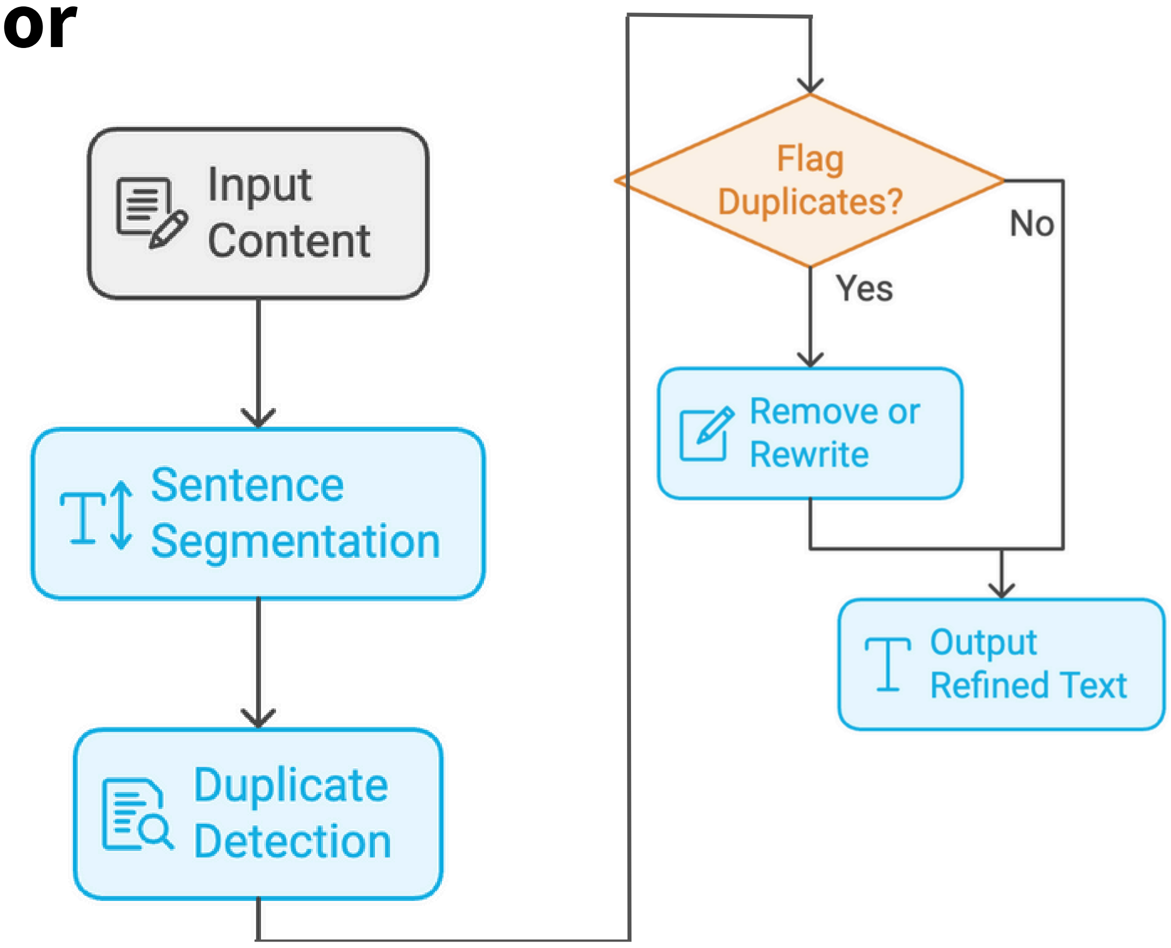
LLM-As-Task Planner

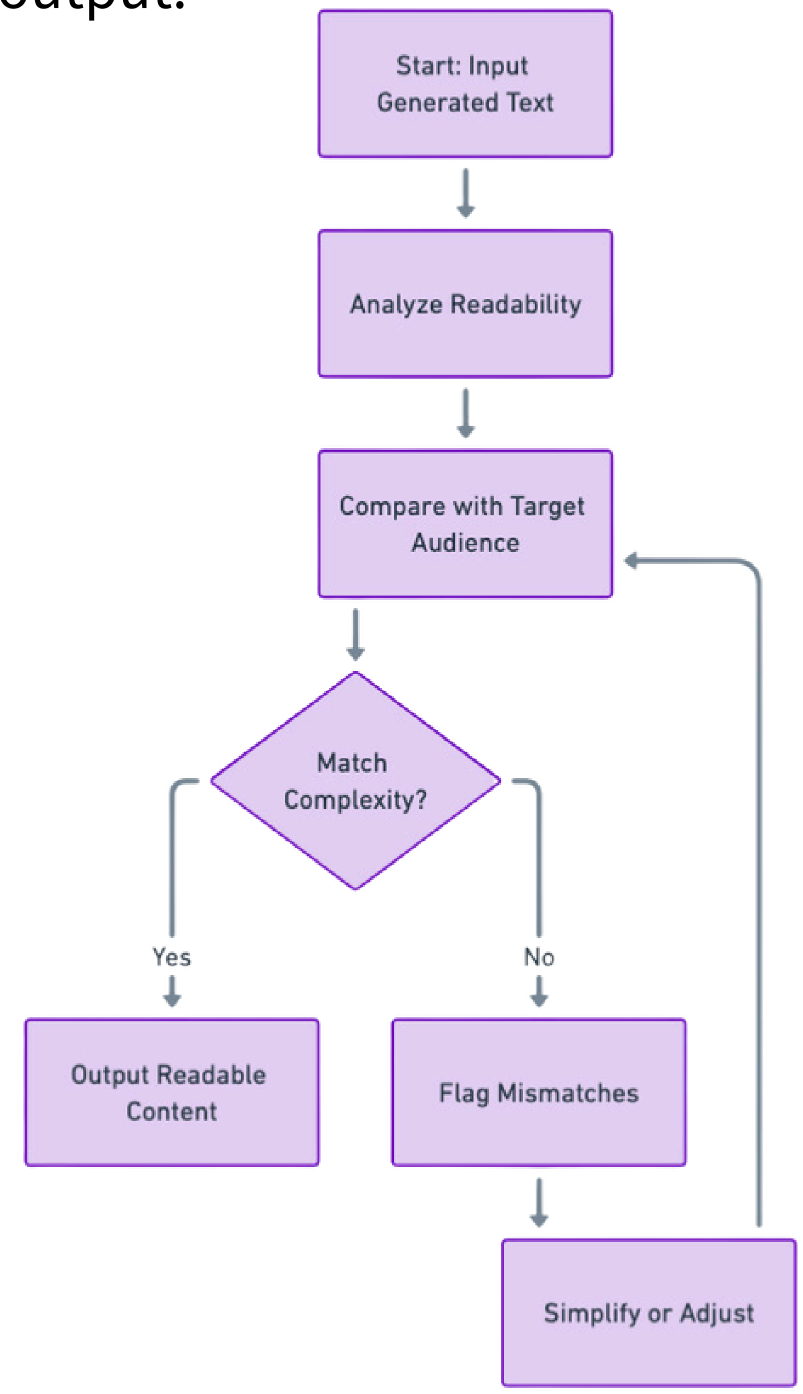
Credit: generalcognitions

Language Quality Guardrails

11. Duplicate sentence eliminator

- Spots Repeated Lines: Detects sentences that are unnecessarily repeated.
- Removes Redundancy: Deletes duplicates to make responses concise.
- Improves Clarity: Makes content easier to read and understand.
- Keeps Answers Focused: Ensures no extra fluff in the output.





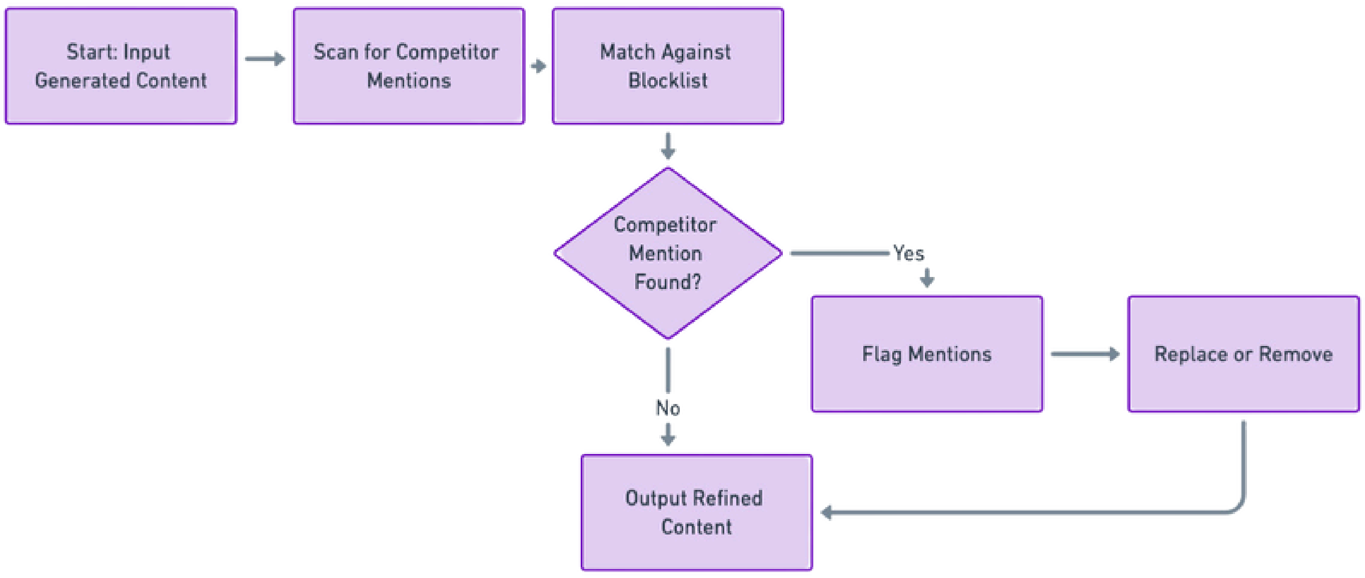
12. Readability Level Evaluator

- Checks Text Complexity: Ensures the content matches the reader’s skill level.
- Uses Smart Tools: Assesses readability with algorithms like Flesch-Kincaid.
- Simplifies When Needed: Adjusts text to be clear for beginners or experts.
- Enhances Understanding: Makes sure all users can grasp the content easily.

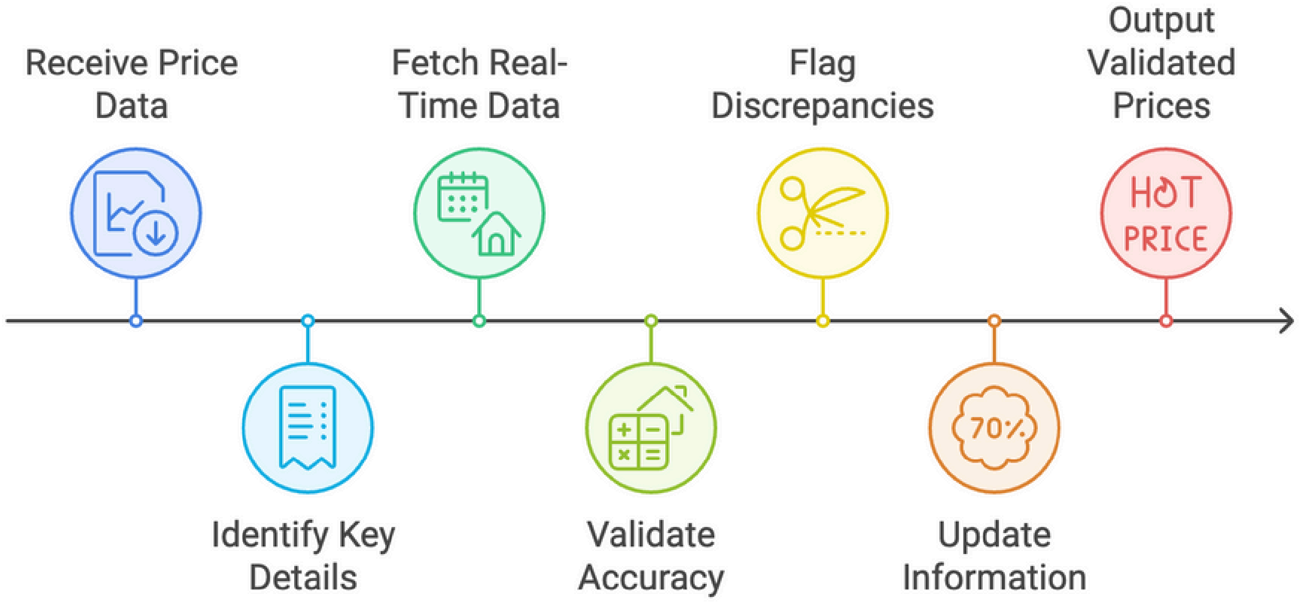
Content Validation and Integrity Guardrails

13. Competitor mention blocker

- Detects Rival Mentions: Spots references to competitor brands in text.
- Neutralizes Content: Replaces or removes competitor names.
- Keeps Focus on You: Ensures responses highlight your brand only.
- Supports Business Goals: Prevents unintentional promotion of rivals.


Start: Input Generated Content
Scan for Competitor Mentions
Match Against Blocklist
Competitor Mention Found?
Yes
Flag Mentions
Replace or Remove
No
Output Refined Content

14. Price Quote Validator

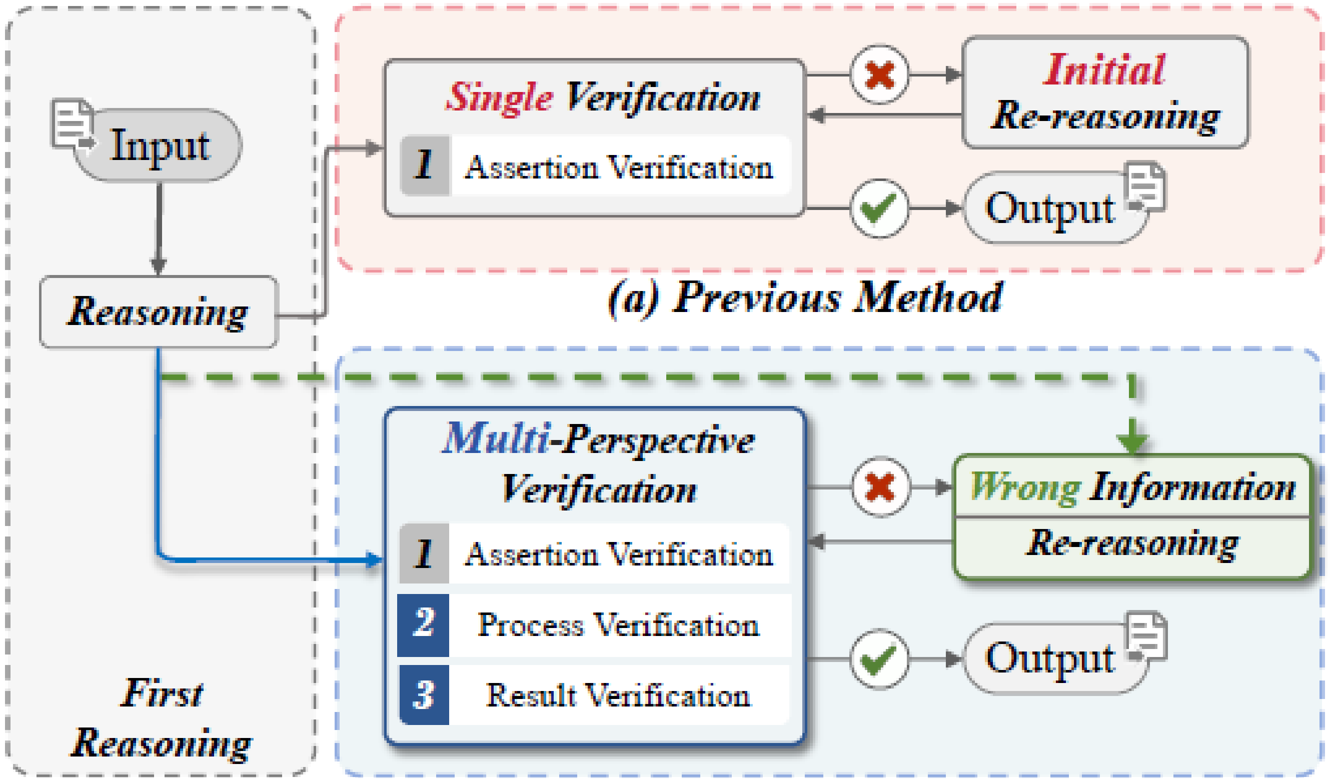


- Checks Pricing Accuracy: Verifies price details in responses with real-time data.
- Uses Trusted Sources: Cross-references prices with reliable databases.
- Corrects Mistakes: Fixes any incorrect or outdated price information.
- Builds Trust: Ensures users get accurate and reliable pricing details

Content Validation and Integrity Guardrails

15. Source Context Verifier

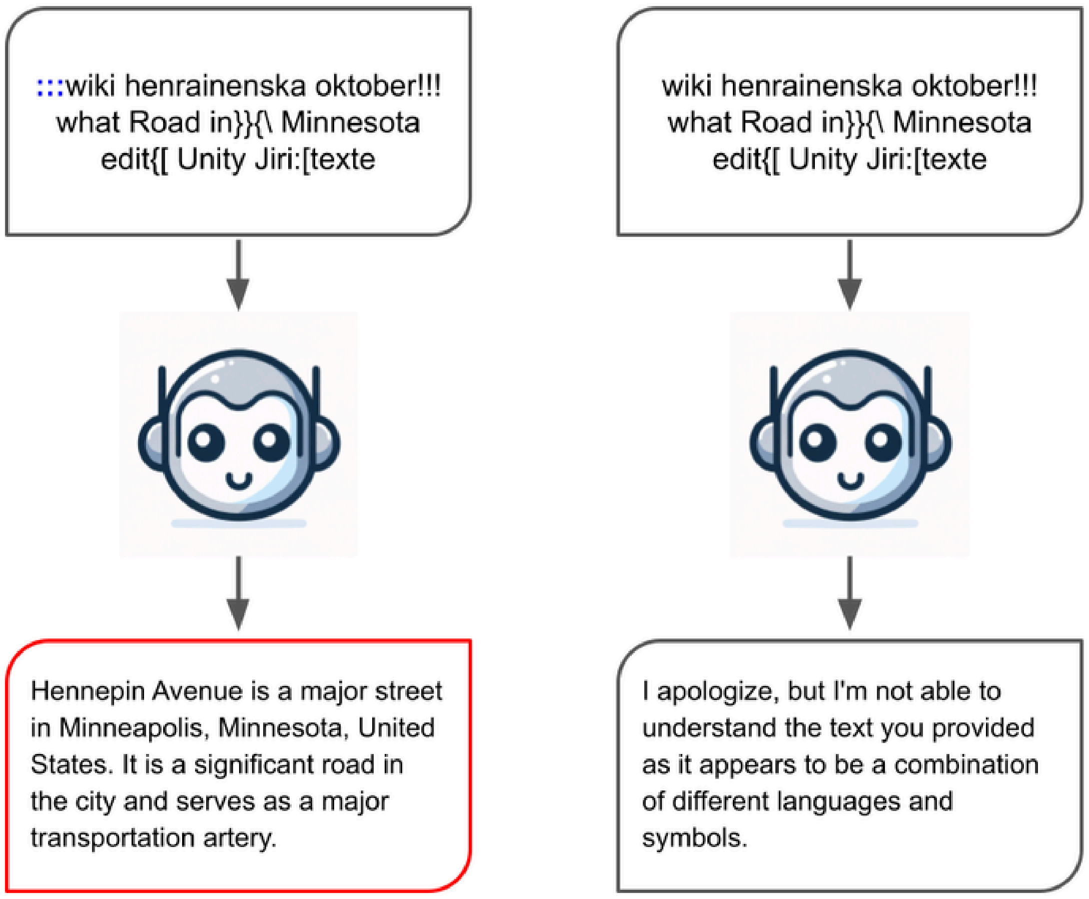
- Checks Facts: Ensures quotes and references match the original source.
- Prevents Misrepresentation: Corrects any misinterpreted information.
- Cross-References Material: Verifies details with trusted external sources.
- Keeps Content Accurate: Stops the spread of false or misleading info.



Credit: medium

16. Gibberish Content Filter

- Spots Nonsense: Detects outputs that are illogical or incoherent.
- Analyzes Sentence Structure: Ensures responses make logical sense.
- Removes Jumbled Text: Filters out meaningless or random content.
- Ensures Clarity: Guarantees all responses are clear and understandable.



:::wiki henrainenska oktober!!! what Road in}}{\ Minnesota edit{[Unity Jiri:[texte

Hennepin Avenue is a major street in Minneapolis, Minnesota, United States. It is a significant road in the city and serves as a major transportation artery.

wiki henrainenska oktober!!! what Road in}}{\ Minnesota edit{[Unity Jiri:[texte

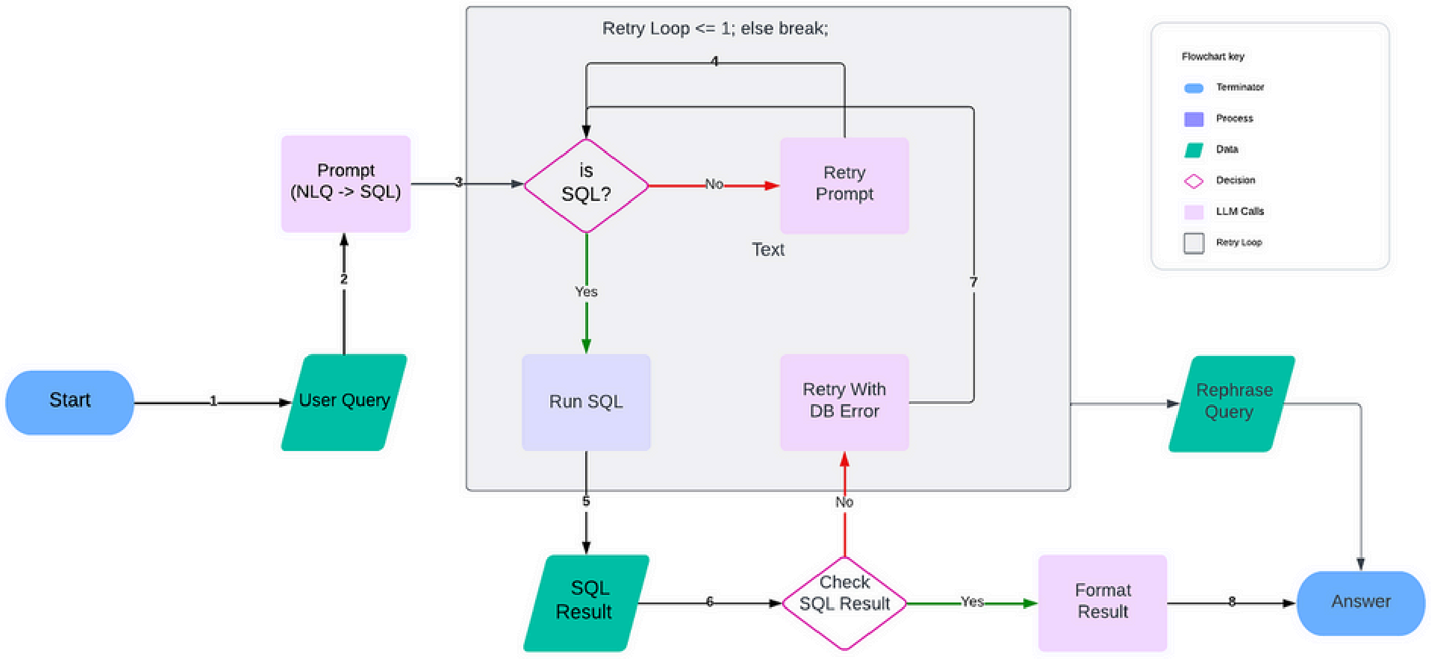
I apologize, but I'm not able to understand the text you provided as it appears to be a combination of different languages and symbols.

Credit: arxiv

Logic and Functionality Validation Guardrails

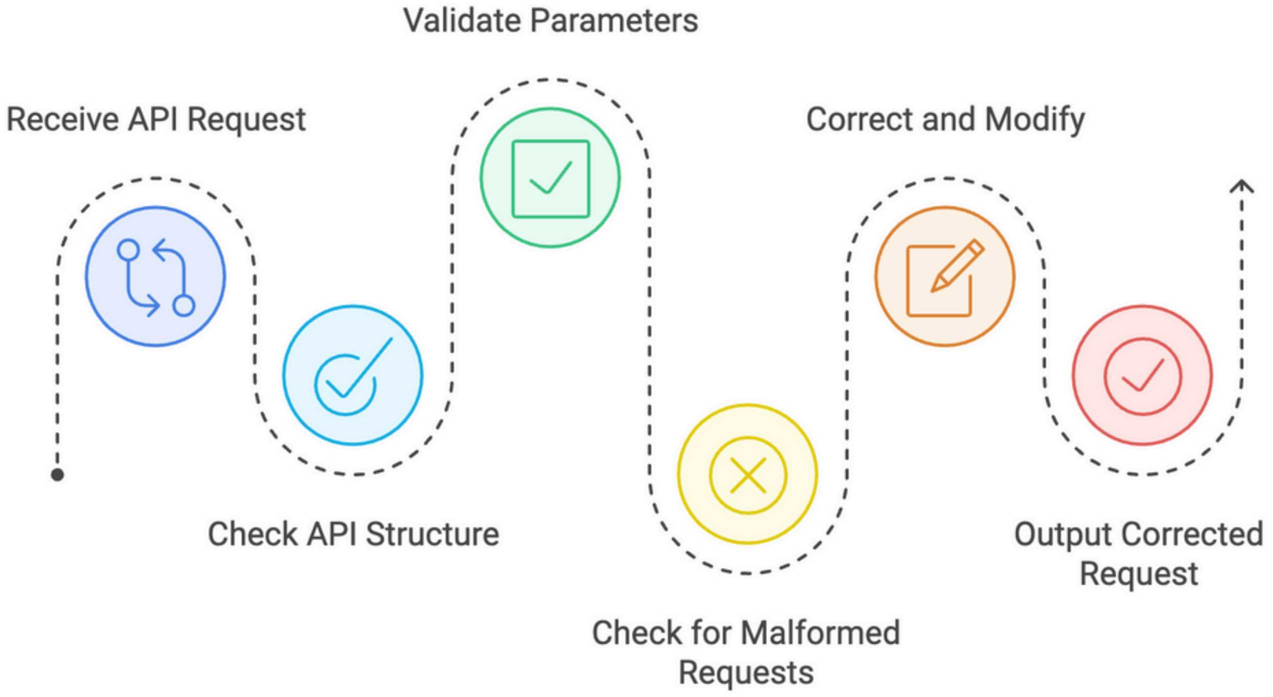
17. SQL Query Validator

- Checks Syntax: Ensures SQL queries are correctly written.
- Prevents Errors: Flags and fixes any mistakes in the query.
- Ensures Safety: Protects against security risks like SQL injection.
- Validates Queries: Confirms the query can run safely and correctly.



credit: medium

18. OpenAPI Specification Checker

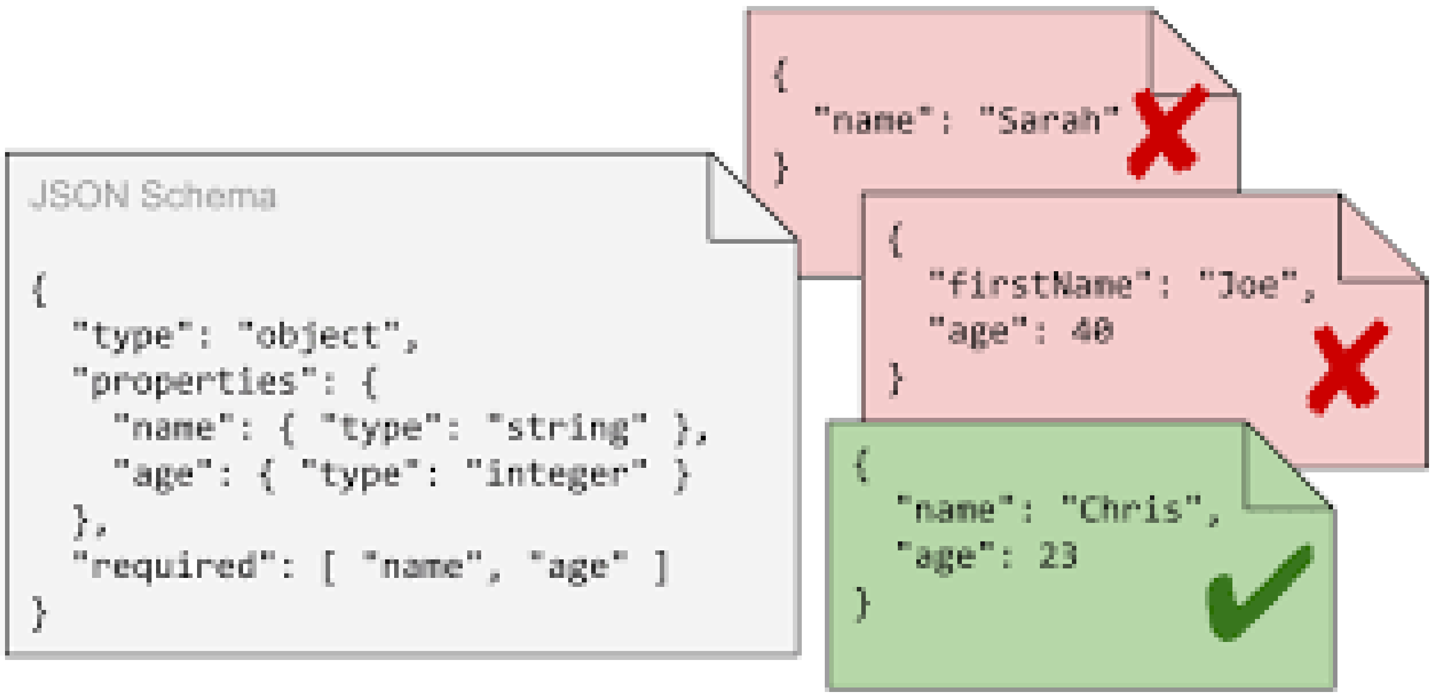


- Validates API Calls: Ensures API requests follow proper formats.
- Checks Parameters: Flags missing or incorrect parameters.
- Corrects Structure: Fixes any issues to meet OpenAPI standards.
- Ensures Functionality: Ensures API calls work as intended.

Logic and Functionality Validation Guardrails

19. JSON Format Validator

- Checks JSON Structure: Ensures JSON data is correctly formatted.
- Fixes Errors: Corrects missing or wrong keys and values.
- Prevents Mistakes: Ensures smooth data exchange in applications.
- Validates Schema: Verifies that the JSON follows the right structure.



JSON Schema

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "age": { "type": "integer" }
  },
  "required": [ "name", "age" ]
}
```

```
{
  "name": "Sarah"
}
```

✗

```
{
  "firstName": "Joe",
  "age": 40
}
```

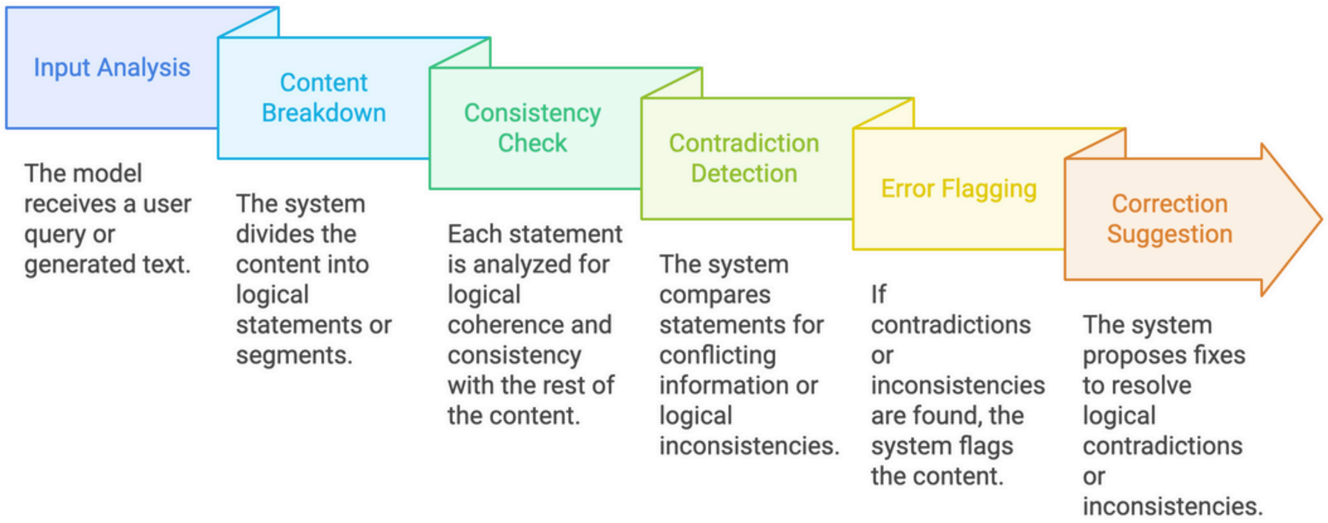
✗

```
{
  "name": "Chris",
  "age": 23
}
```

✓

Credit: JSON Editor

20. Logical Consistency Checker



Input Analysis

The model receives a user query or generated text.

Content Breakdown

The system divides the content into logical statements or segments.

Consistency Check

Each statement is analyzed for logical coherence and consistency with the rest of the content.

Contradiction Detection

The system compares statements for conflicting information or logical inconsistencies.

Error Flagging

If contradictions or inconsistencies are found, the system flags the content.

Correction Suggestion

The system proposes fixes to resolve logical contradictions or inconsistencies.

- Detects Contradictions: Spots any logical errors in the response.
- Ensures Consistency: Makes sure all statements align with each other.
- Analyzes Flow: Checks if the response makes sense overall.
- Corrects Inconsistencies: Fixes any contradictory or illogical content.

Follow to stay updated on AI/ML

👍 LIKE

💬 COMMENT

🔁 REPOST