

Stop the Loop That Hallucinates and Burns Tokens

Two symptoms, one ungoverned loop. A better prompt fixes neither.

The fix lives in the harness around the model.





Elizabeth Fuentes Leone

Developer Advocate @ AWS

- MS in Data Science
- 107+ technical articles on dev.to
- Bilingual: English / Spanish

Resources



bit.ly/4exJLGX



Two Symptoms, One Loop

It hallucinates

- Invents a value when it finds no data
- Picks the wrong tool when two look alike
- Ignores the rule you set

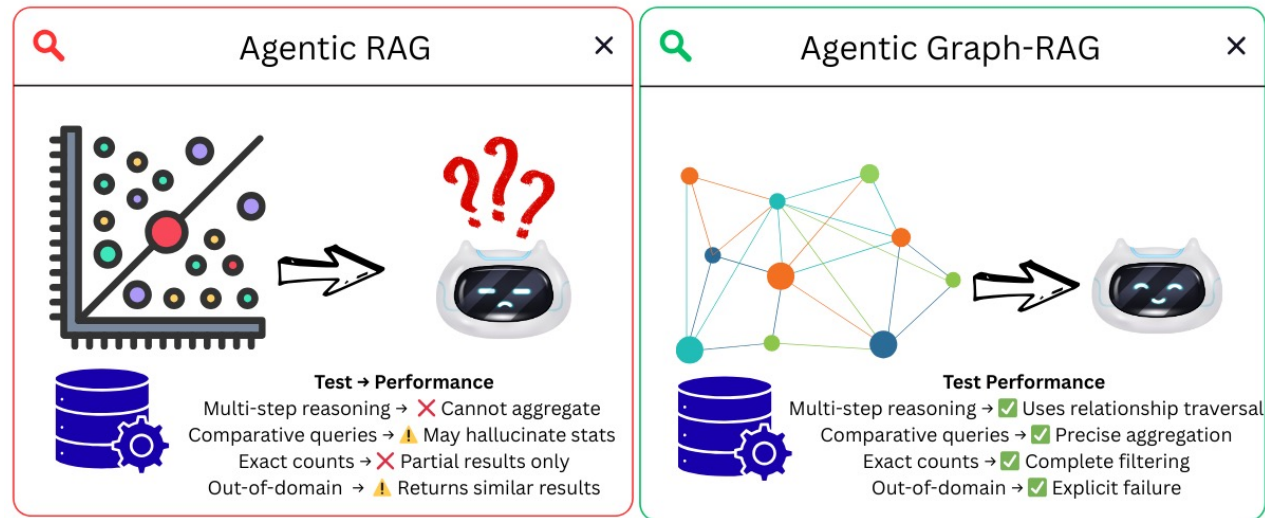
It burns tokens

- Ships every tool description on every call
- Retries the same call 14 times on vague feedback
- Overflows the window with a giant tool output

1 · Ground the Answer

RAG guesses. Graph-RAG computes.

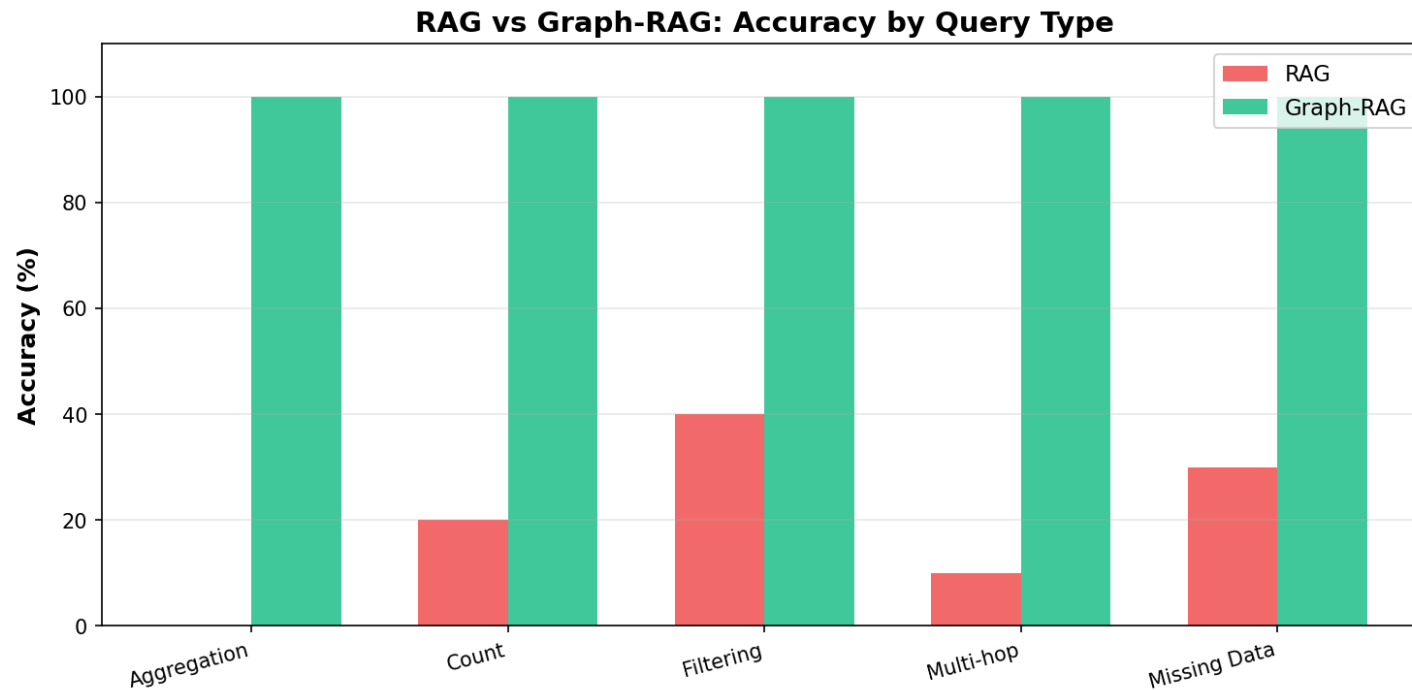
Text retrieval returns chunks and the agent fabricates aggregations. A structured source computes the exact result.



1 · Ground the Answer

Query the graph, return real values

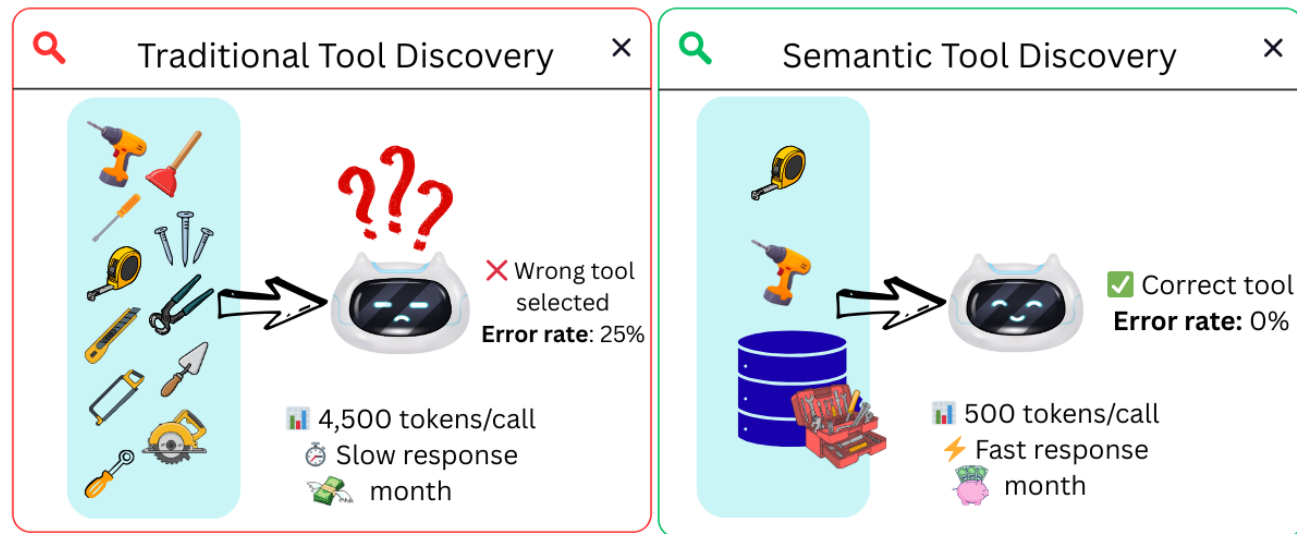
Accuracy by query type, measured in the demo across 300 hotel FAQ documents.



2 · Route Tools by Meaning

Every tool, on every call

A crowded tool list raises wrong picks AND token cost. This is where the two symptoms meet.



2 · Route Tools by Meaning

Hand the model only the few that fit

1,450

tokens/query · 29 tools



150

tokens/query · top 3 tools

Fewer tokens per call, and a shorter list means fewer wrong picks.

```
[1/3] Traditional - 29 tools every query...
What's the weather in Paris?      | 1800 tokens (est)
Find flights from NYC to London   | 1800 tokens (est)
Book a hotel in Rome for John     | 1600 tokens (est)
```

```
[2/3] Semantic - Top-3 tools per query...
What's the weather in Paris?      | 500 tokens (est)
Find flights from NYC to London   | 500 tokens (est)
Book a hotel in Rome for John     | 300 tokens (est)
```

```
[3/3] Semantic + Memory - Single agent, swap tools...
What's the weather in Paris?      | 500 tokens (est)
Find flights from NYC to London   | 900 tokens (est)
Book a hotel in Rome for John     | 1300 tokens (est)
```

RESULTS

```
Total tokens:
Traditional:   5200 tokens
Semantic:      1300 tokens (+75.0%)
Semantic+Memory: 2700 tokens (+48.1%)
```

Query	Trad	Sem	Mem	Saved
What's the weather in Paris?	1800	500	500	1300
Find flights from NYC to London	1800	500	900	900
Book a hotel in Rome for John	1600	300	1300	300

✓ Key Finding:

- Traditional sends 29 tools every query
- Semantic sends only 3 tools per query
- Semantic saves 3900 tokens (75.0%) vs Traditional
- Memory approach accumulates conversation history
- Memory uses 1400 MORE tokens than Semantic (conversation context)
- But still saves 2500 tokens (48.1%) vs Traditional

3 · Rules It Cannot Bypass

Prompt rules get reasoned around

A rule written only in natural language is a suggestion. The model talks its way past it.



Prompt Engineering



```
system_prompt = ""
IMPORTANT: Never confirm
without payment

CRITICAL: Max 10 guests
""
```

- ✓ Natural Language
- ✓ Flexible
- ✓ Conversational
- ✗ Can hallucinate



- ✗ LLM can ignore
- ⚠ Not enforced



Symbolic Rules



```
if not payment_verified:
    return "BLOCKED"

if guests > 10:
    return "BLOCKED"
```

- ✓ Business Logic
- ✓ Deterministic
- ✓ Verifiable



- ✓ Enforced at execution
- 🛡 Cannot bypass



© 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3 · Block or Steer

Hard-block what's forbidden, steer the rest

Hooks cancel a forbidden call before it runs. Steer messages let the agent self-correct: 15 guests becomes a booking for 10 with a note.

 Hooks (Block) 



What happens:

book_hotel(guests=15)	→ ❌ BLOCKED: exceeds max 10
Agent reports failure	→ ❌ "Would you like to adjust?"
User must respond	→ ❌ Flow interrupted
Booking NOT completed	→ ❌ User frustrated

 Agent Control (Self-Correct) 



What happens:

book_hotel(guests=15)	→ ✅ Guide("reduce to 10")
Agent retries with 10 guests	→ ✅ Automatic correction
"Adjusted to 10, BK002"	→ ✅ User informed
Booking COMPLETED	→ ✅ Flow uninterrupted

4 · Stop the Burn

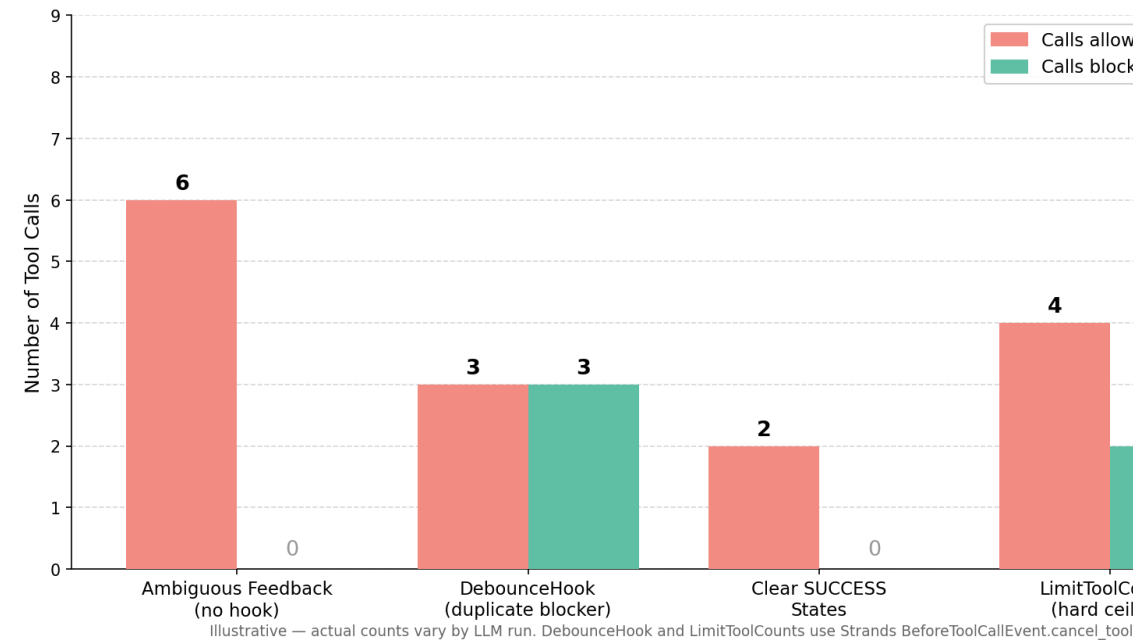
Kill the runaway loop

Ambiguous feedback makes the agent retry. A clear done signal plus duplicate blocking stops it.

14 → 2

tool calls for one booking · 7x fewer

Tool Calls: Allowed vs Blocked by Strategy



4 · Stop the Burn

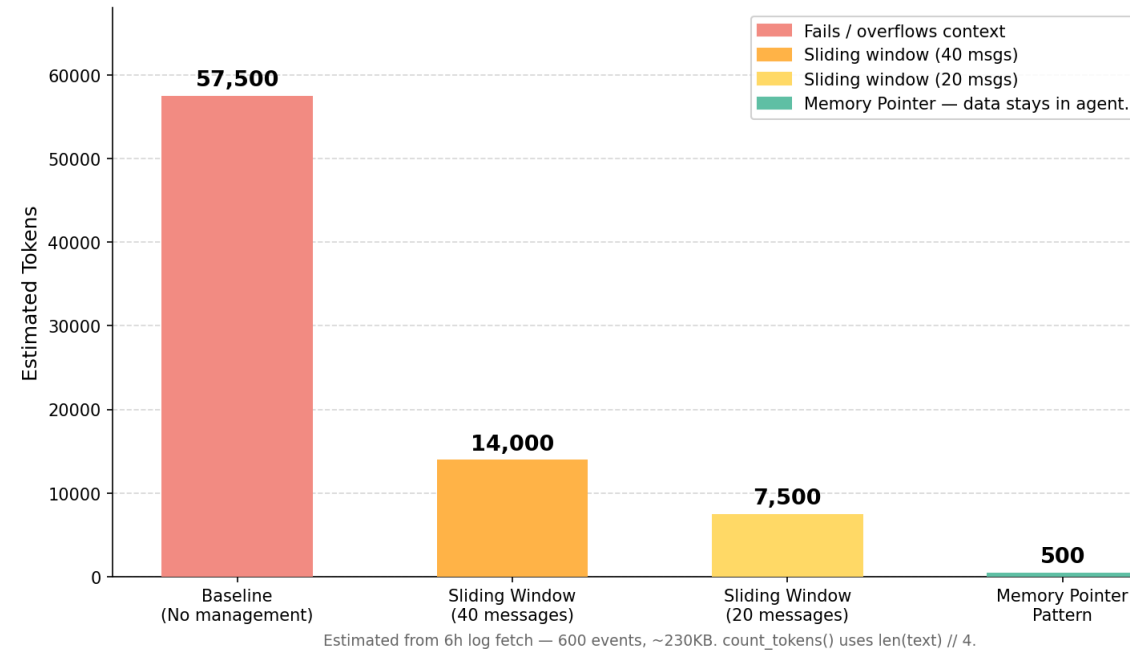
Keep big outputs out of the window

Store the large tool output outside context, recall it by exact reference.

~97%

fewer tokens in context · ~18–20K → ~490, same query

Token Usage by Context Strategy



Demo Time

Every control, live on the same booking agent — before and after.



What You Take Home

- Map each failure to the control that fixes it, and where in the loop it belongs
- Ground answers and route tools by meaning to cut fabrications and token cost together
- Block what's forbidden, steer the rest, and stop runaway loops and overflow

Thank You!

Elizabeth Fuentes Leone

Developer Advocate @ AWS

elifuentes.tech



Resources

bit.ly/4exJLGX



Blog & Socials

elifuentes.tech



© 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.