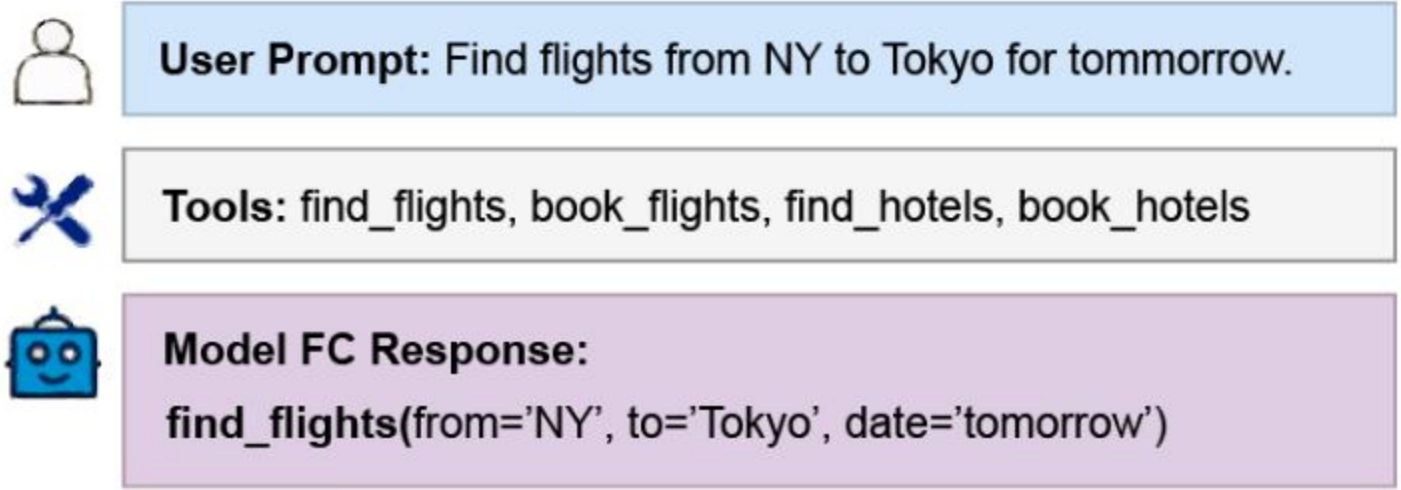
Agent Evals Literature Review

Ran Tavory @TII
Sept 2025

Example 1



User Prompt: Find flights from NY to Tokyo for tommorrow.

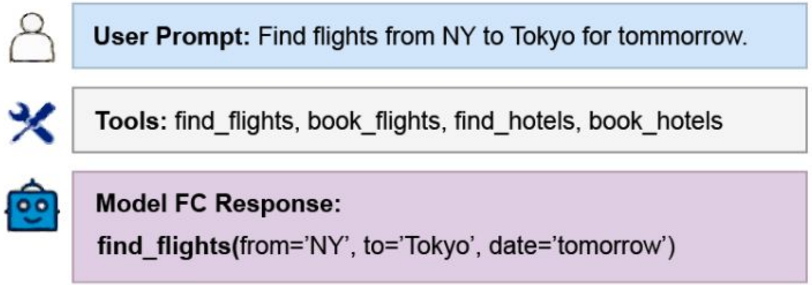
Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

find_flights(from=’NY’, to=’Tokyo’, date=’tomorrow’)

Example 1

- Did the model invoke the right function?
- Did it pass the correct parameters?



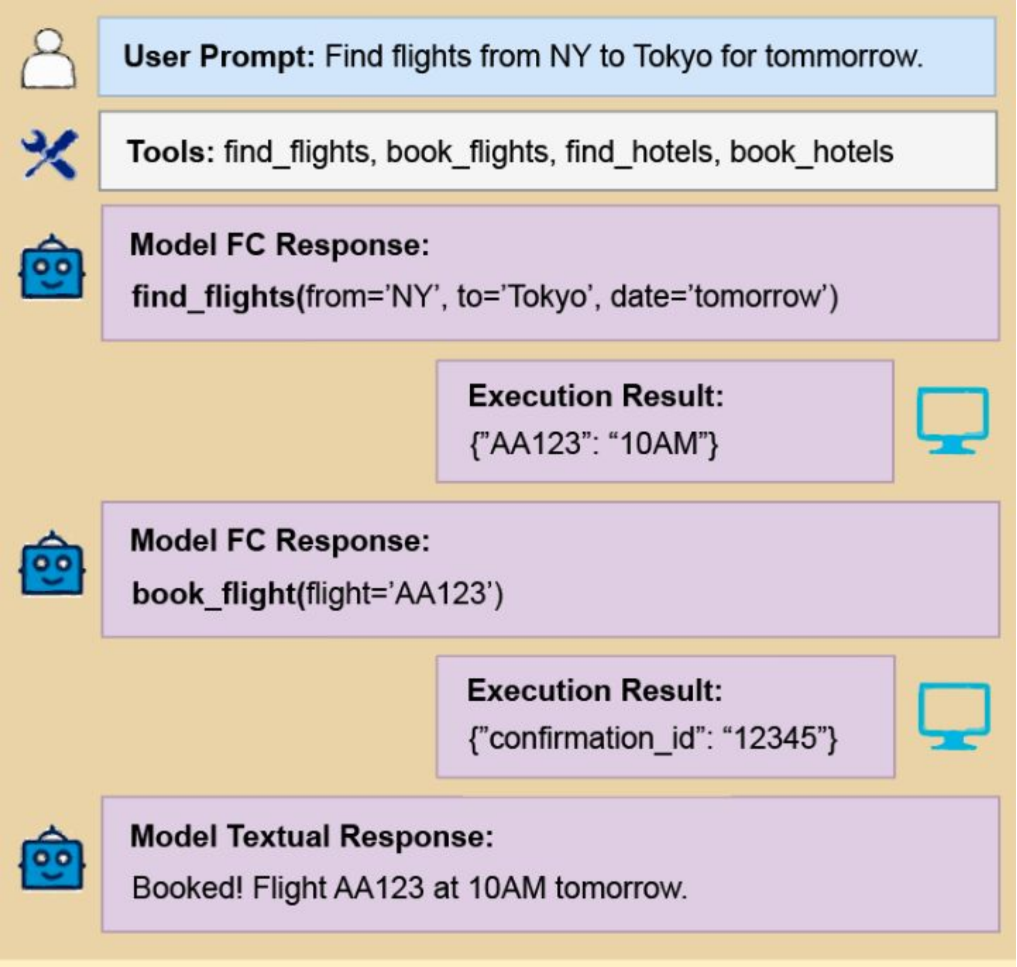
User Prompt: Find flights from NY to Tokyo for tommorrow.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

find_flights(from='NY', to='Tokyo', date='tomorrow')

Example 2



User Prompt: Find flights from NY to Tokyo for tommorrow.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

find_flights(from=’NY’, to=’Tokyo’, date=’tomorrow’)

Execution Result:

{”AA123”: “10AM”}

Model FC Response:

book_flight(flight=’AA123’)

Execution Result:

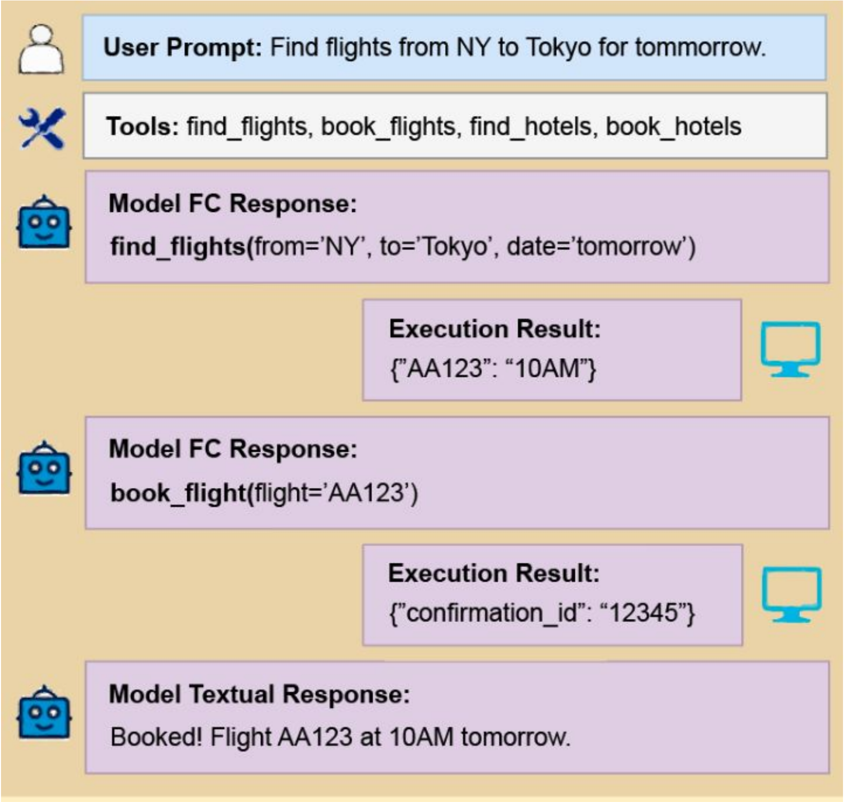
{”confirmation_id”: “12345”}

Model Textual Response:

Booked! Flight AA123 at 10AM tomorrow.

Example 2

- Did the model invoke the right function?
- Did the model invoke the right function**s**?
- Did it pass the correct parameters?
- Did it interpret the result correctly?
- Did it invoke the functions in the correct order?
  - There are multiple correct orders
- Did it return a correct and useful answer to the user?
- Did it produce a side-effect? (book a flight). Was it correct?
  - Did it produce unwanted side-effects?



User Prompt: Find flights from NY to Tokyo for tommorrow.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:
find_flights(from=’NY’, to=’Tokyo’, date=’tomorrow’)

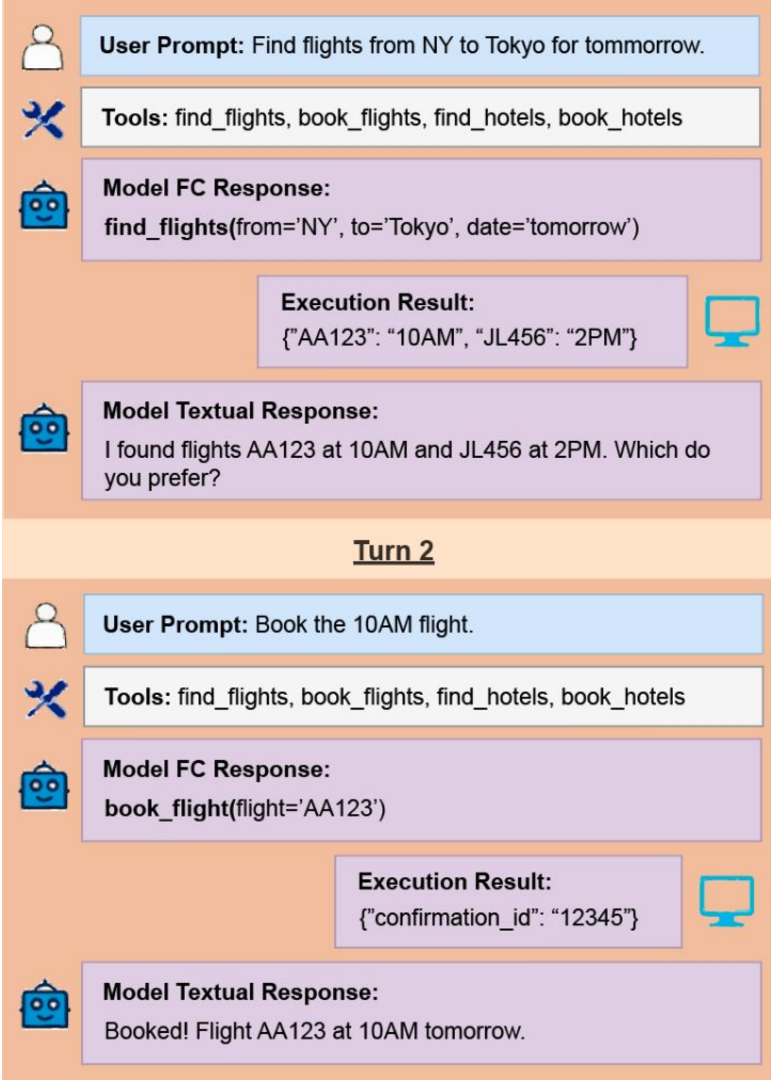
Execution Result:
{”AA123”: “10AM”}

Model FC Response:
book_flight(flight=’AA123’)

Execution Result:
{”confirmation_id”: “12345”}

Model Textual Response:
Booked! Flight AA123 at 10AM tomorrow.

Example 3



User Prompt: Find flights from NY to Tokyo for tommorrow.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

find_flights(from=’NY’, to=’Tokyo’, date=’tomorrow’)

Execution Result:

{”AA123”: “10AM”, “JL456”: “2PM”}

Model Textual Response:

I found flights AA123 at 10AM and JL456 at 2PM. Which do you prefer?

Turn 2

User Prompt: Book the 10AM flight.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

book_flight(flight=’AA123’)

Execution Result:

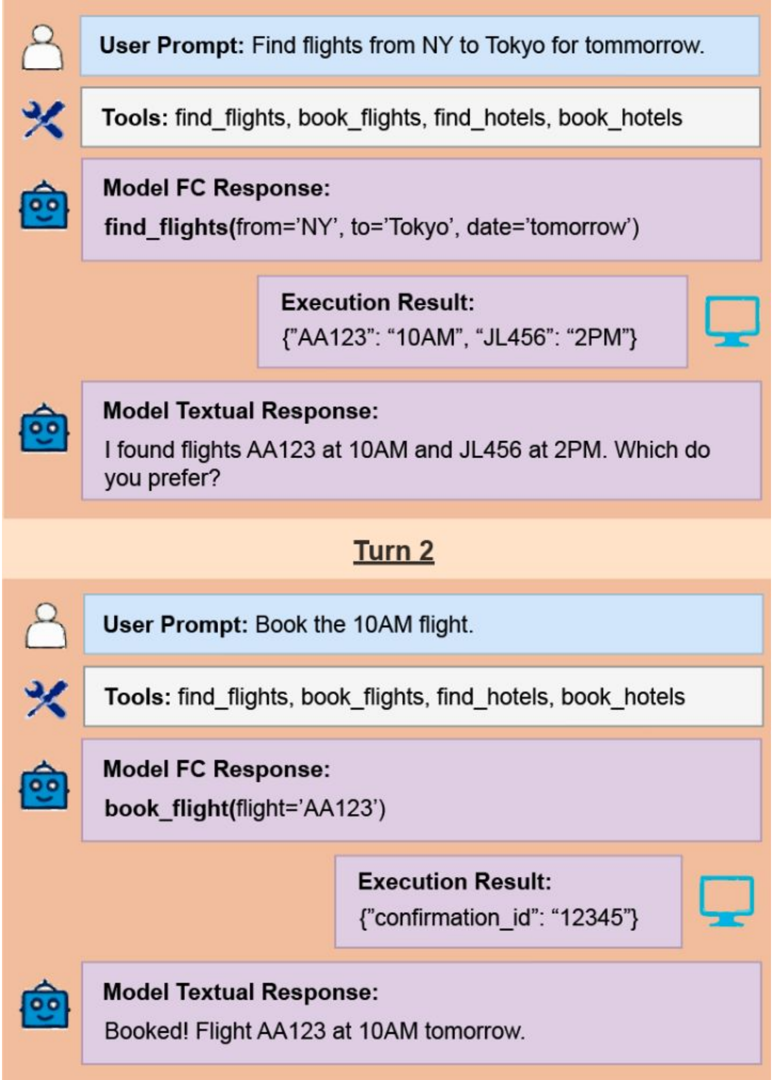
{”confirmation_id”: “12345”}

Model Textual Response:

Booked! Flight AA123 at 10AM tomorrow.

Example 3

- Did the model invoke the right functions?
- Did it pass the correct parameters?
- Did it interpret the result correctly?
- Did it invoke the functions in a reasonable oder?
- Did it return a correct and useful answer to the user?
- Did it make a side-effect? (book a flight). Was it correct?
  - Did it call unwanted side-effects?
- Did it interpret the user meaning(s) correctly?
- Did it succeed in all turns?
- Did it reach a final goal intended by the user?



User Prompt: Find flights from NY to Tokyo for tommorrow.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

find_flights(from='NY', to='Tokyo', date='tomorrow')

Execution Result:

{"AA123": "10AM", "JL456": "2PM"}

Model Textual Response:

I found flights AA123 at 10AM and JL456 at 2PM. Which do you prefer?

Turn 2

User Prompt: Book the 10AM flight.

Tools: find_flights, book_flights, find_hotels, book_hotels

Model FC Response:

book_flight(flight='AA123')

Execution Result:

{"confirmation_id": "12345"}

Model Textual Response:

Booked! Flight AA123 at 10AM tomorrow.

Example n...

- Did the model invoke the right functions?
- Did it pass the correct parameters?
- Did it interpret the result correctly?
- Did it invoke the functions in a reasonable oder?
- Did it return a correct and useful answer to the user?
- Did it succeed in all turns?
- Did it reach a final goal intended by the user?
- Did it make any side-effects? (book a flight). Are they correct?
    - Did it cause unwanted side-effects?

- Did it achieve the goal in a reasonable number of steps?
- And reasonable number of tokens?
- Did it leak personal data?
- Did it invoke unsecure methods?
- Did it repeatedly and consistently succeed?
- ...

Topics:

- Benchmarks overview
- Aspects measured
- Metering methods
- Metrics

Benchmarks overview

- Berkeley Function-Calling Leaderboard (BFCL) https://gorilla.cs.berkeley.edu/
    - Comprehensive, various functional aspects. COMPONENT, E2E.
- τ-bench https://arxiv.org/abs/2406.12045
    - Simulated environments e.g. Flight Reservation, Retail. E2E.
- ComplexFuncBench https://arxiv.org/abs/2410.12952
    - Complex function-calling capabilities COMPONENT, E2E.
- WebArena https://arxiv.org/abs/2307.13854
    - Full web-browsing agents. E2E.
- OPENAGENTSAFETY https://arxiv.org/abs/2507.06134 and HAICOSYSTEM-EVAL https://arxiv.org/abs/2409.16427
    - Safety and performance of AI agents COMPONENT

Benchmarks citations

- Berkeley Function-Calling Leaderboard (BFCL) https://gorilla.cs.berkeley.edu/
  - Patil, S. G., Mao, H., Ji, C. C.-J., Yan, F., Suresh, V., Stoica, I., & Gonzalez, J. E. (2025). The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. Forty-second International Conference on Machine Learning. (An earlier version cites the year 2024), **Berkeley**.
- τ-bench https://arxiv.org/abs/2406.12045
  - Yao, S., Shinn, N., Razavi, P., & Narasimhan, K. (2024). tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv preprint arXiv:2406.12045, **Sierra**
- ComplexFuncBench https://arxiv.org/abs/2410.12952
  - Zhong, L., Du, Z., Zhang, X., Hu, H., & Tang, J. (2024). ComplexFuncBench: Exploring Multi-Step and Constrained Function Calling under Long-Context Scenario. arXiv preprint arXiv:2410.12952, **Baichuan Inc.**, **Peking University**
- WebArena https://arxiv.org/abs/2307.13854
  - Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al. (2023). Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854, **Carnegie Mellon University**
- OPENAGENTSAFETY https://arxiv.org/abs/2507.06134
  - Vijayvargiya, S., Soni, A. B., Zhou, X., Wang, Z. Z., Dziri, N., Neubig, G., & Sap, M. (2025). OPENAGENTSAFETY: A Comprehensive Framework for Evaluating Real-World AI Agent Safety. arXiv preprint arXiv:2507.06134, **Language Technologies Institute, Carnegie Mellon University, Allen Institute for Artificial Intelligence**
- HAICOSYSTEM-EVAL https://arxiv.org/abs/2409.16427
  - Zhou, X., Kim, H., Brahman, F., Jiang, L., Zhu, H., Lu, X., Xu, F., Lin, B. Y., Choi, Y., Mireshghallah, N., Le Bras, R., & Sap, M. (2024). HAICOSYSTEM: An Ecosystem for Sandboxing Safety Risks in Human-AI Interactions. arXiv preprint arXiv:2409.16427. **Language Technologies Institute, Carnegie Mellon University, Allen Institute for AI**

COMPONENT v/s E2E.

- COMPONENT level v/s E2E metrics
- More on that later.

Aspects measured and focus

- BFCL
  - Aspects: parallel and multiple function calls, relevance detection (hallucinations).
  - Focus:
    - V1: single-turn, simple, parallel, and multiple function calling
    - V2: Real-world scenarios, data contamination and bias.
    - V3: Multi-turn, multi-step, back-and-forth conversations, ambiguous prompts, missing parameters, missing functions, long contexts.
    - V4: Specific tools: Web Search, Memory, Format Sensitivity
- τ-bench
  - Aspects: Adherence to policy guidelines, consistency across multiple trials.
  - Focus: Reliability, consistency following guidelines in specific domains
- ComplexFuncBench
  - Aspects: Multi-step function calling, user constraints, parameter value reasoning, long parameter values, long context lengths
  - Focus: Infer correct parameter values from implicit information, complex multi-step workflows
- WebArena
  - Aspects: Decision-making, long-horizon planning, web navigation, web form-filling
  - Focus: realistic web environment
- OPENAGENTSAFETY and HAICOSYSTEM-EVAL
  - Aspects: safety (8 categories, including legal, privacy, harmful decision-making, data loss, financial loss, unsafe code, ...).
  - Focus: Safety in realistic, high-risk, multi-turn, multi-user scenarios with diverse user intentions (benign and malicious)

Aspects measured and focus

Aspect/Benchmark	BFCL	τ-bench	ComplexFuncBench
Single-turn correctness	AST		
Multiple function calls			
Parallel function calls			

Metering methods

- BFCL V1: A hand-crafted benchmark. single-turn function calls.
    - Metric: **AST**, **Cost**, **Latency**
- BFCL V2: Focus on user-contributed real-world functions documentation and queries.
    - Metric: **AST**, **non-invocation**
- BFCL V3 (Multi-Turn & Multi-Step): Complex scenarios. Missing parameters, long contexts.
    - Metric: **Desired State**, **Execution paths**
- BFCL V4: Focus on 1) web search 2) Memory usage and 3) Sensitivity to variations in input and output formats
    - Metric: **AST**, specific **Desired State**

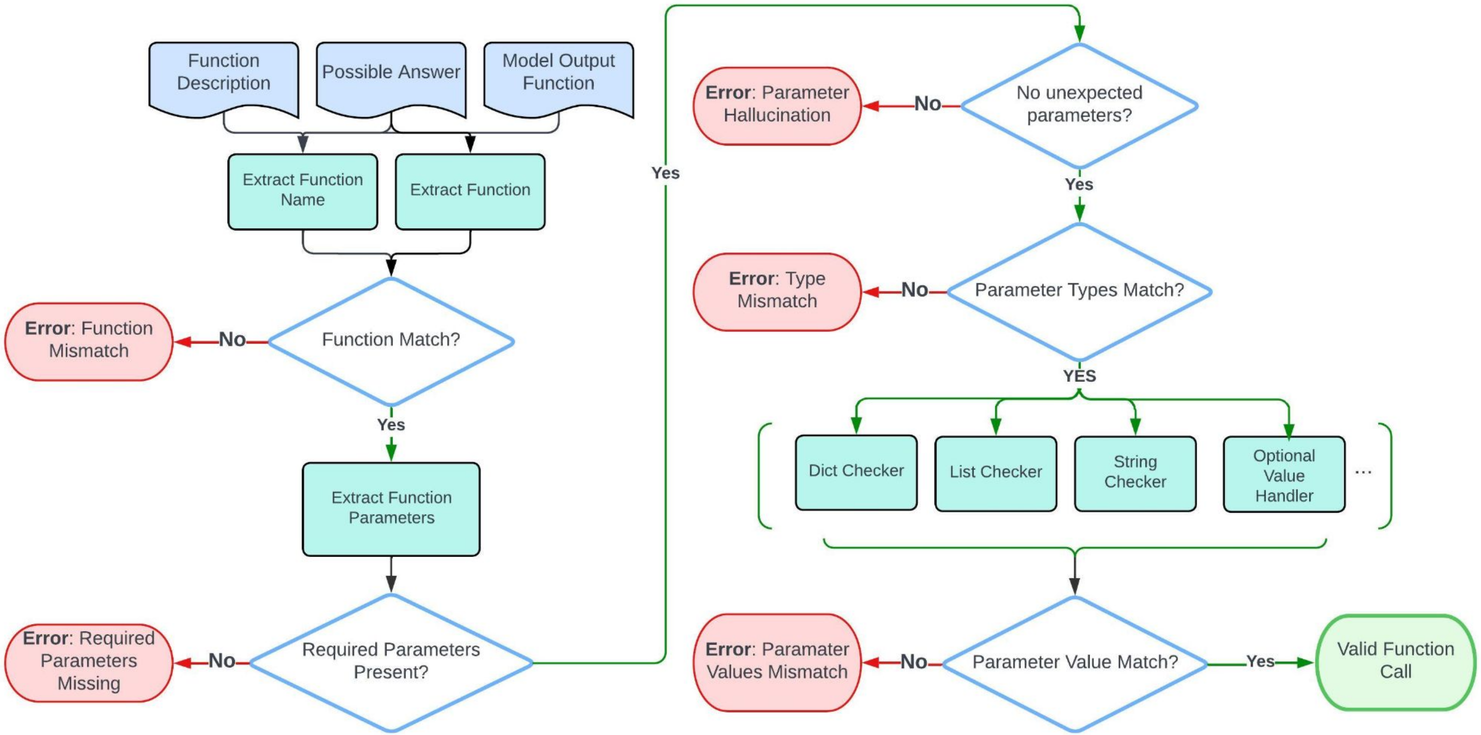
Metering methods

- τ-bench: Emulates a simulated user and agent with stub API tools and policy guidelines
  - Metric: **Desired State**. pass^k
- ComplexFuncBench: real-world scenarios (from domains like Booking.com)
  - Metric: A matching approach (rule-based, response-based, and LLM-based) measuring **Success Rate** and **Call Accuracy**
- WebArena: Web simulator, long-horizon planning, navigating hyperlinks, filling forms
  - Metrics: Verifying correct sequence of operations. Task completion rate.

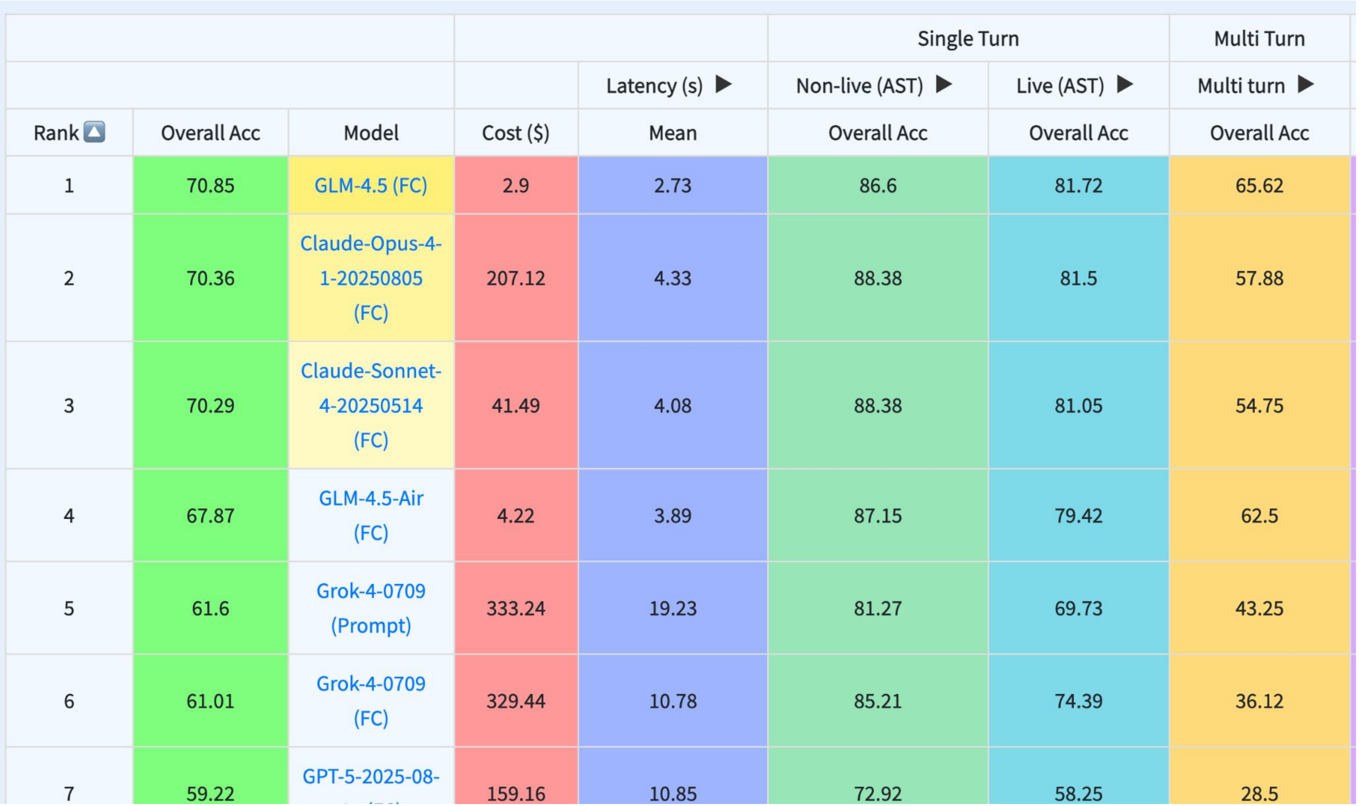
Discussion - What do we care about

- Correctness
  - Correct function invocation COMPONENT.
  - Correct parameters COMPONENT.
    - Simple parameters such as short strings and ints
    - Complex parameters such as search queries
  - Refusal to hallucinations COMPONENT.
  - User intent goal achieved E2E.
    - Desired state
    - Desired output
- Reliability / Repeatability COMPONENT, E2E.
  - E.g. pass^k
- Efficiency COMPONENT, E2E.
  - Cost, Latency, number of turns to complete, number of tokens to complete
- Safety COMPONENT, E2E.
- Security COMPONENT.

AST metric calculation

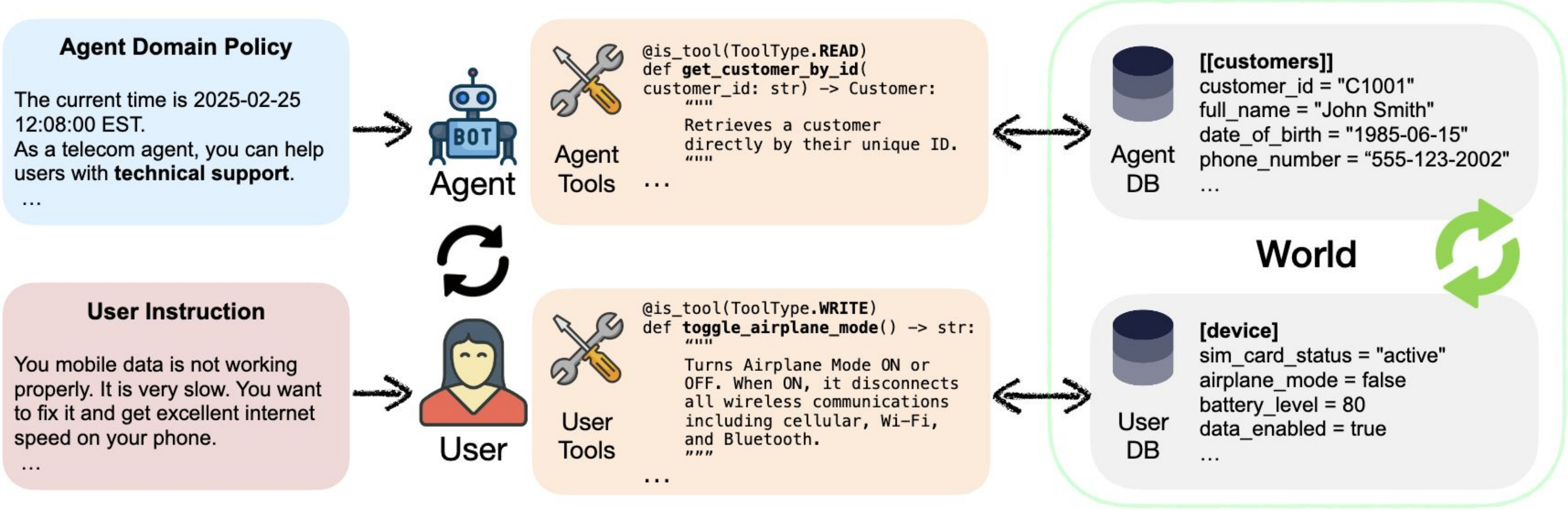


Example: BFCL



					Single Turn		Multi Turn
				Latency (s) ▶	Non-live (AST) ▶	Live (AST) ▶	Multi turn ▶
Rank ▲	Overall Acc	Model	Cost ($)	Mean	Overall Acc	Overall Acc	Overall Acc
1	70.85	GLM-4.5 (FC)	2.9	2.73	86.6	81.72	65.62
2	70.36	Claude-Opus-4-1-20250805 (FC)	207.12	4.33	88.38	81.5	57.88
3	70.29	Claude-Sonnet-4-20250514 (FC)	41.49	4.08	88.38	81.05	54.75
4	67.87	GLM-4.5-Air (FC)	4.22	3.89	87.15	79.42	62.5
5	61.6	Grok-4-0709 (Prompt)	333.24	19.23	81.27	69.73	43.25
6	61.01	Grok-4-0709 (FC)	329.44	10.78	85.21	74.39	36.12
7	59.22	GPT-5-2025-08-	159.16	10.85	72.92	58.25	28.5

Desired State metric calculation (τ-bench)



Agent Domain Policy

The current time is 2025-02-25 12:08:00 EST.
As a telecom agent, you can help users with **technical support**.
…

User Instruction

You mobile data is not working properly. It is very slow. You want to fix it and get excellent internet speed on your phone.
…

```
@is_tool(ToolType.READ)
def get_customer_by_id(
customer_id: str) -> Customer:
    """
    Retrieves a customer
    directly by their unique ID.
    """
...
```

```
@is_tool(ToolType.WRITE)
def toggle_airplane_mode() -> str:
    """
    Turns Airplane Mode ON or
    OFF. When ON, it disconnects
    all wireless communications
    including cellular, Wi-Fi,
    and Bluetooth.
    """
...
```

```
[[customers]]
customer_id = "C1001"
full_name = "John Smith"
date_of_birth = "1985-06-15"
phone_number = “555-123-2002"
...
```

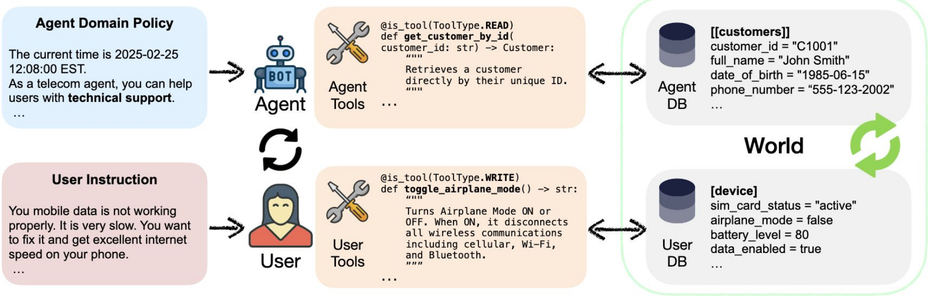
```
[device]
sim_card_status = "active"
airplane_mode = false
battery_level = 80
data_enabled = true
...
```

It requires a lot of work!

Desired State: It requires a lot of work

Kinds of work

1. Effort to invest anyway
    a. The agent
    b. Agent domain policy
    c. Tools implementations
2. Reusable effort
    a. Test harness
    b. User simulator
3. Domain specific effort
    a. Tools simulator
    b. Database simulator



Agent Domain Policy

The current time is 2025-02-25 12:08:00 EST.
As a telecom agent, you can help users with **technical support**.
…

User Instruction

You mobile data is not working properly. It is very slow. You want to fix it and get excellent internet speed on your phone.
…

```
@is_tool(ToolType.READ)
def get_customer_by_id(
customer_id: str) -> Customer:
    """
    Retrieves a customer
    directly by their unique ID.
    """
...
```

```
@is_tool(ToolType.WRITE)
def toggle_airplane_mode() -> str:
    """
    Turns Airplane Mode ON or
    OFF. When ON, it disconnects
    all wireless communications
    including cellular, Wi-Fi,
    and Bluetooth.
    """
...
```

[[customers]]
customer_id = "C1001"
full_name = "John Smith"
date_of_birth = "1985-06-15"
phone_number = “555-123-2002"
…

[device]
sim_card_status = "active"
airplane_mode = false
battery_level = 80
data_enabled = true
…

Pass^k meta-metric (τ-bench)

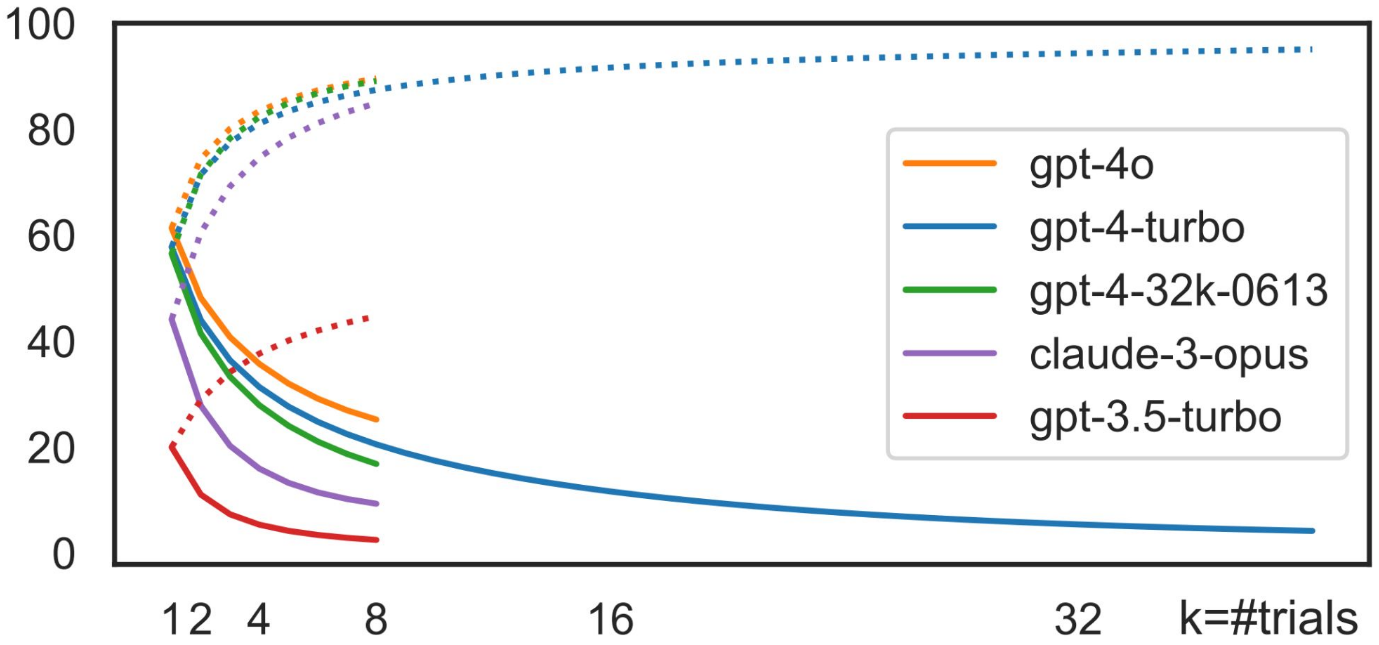
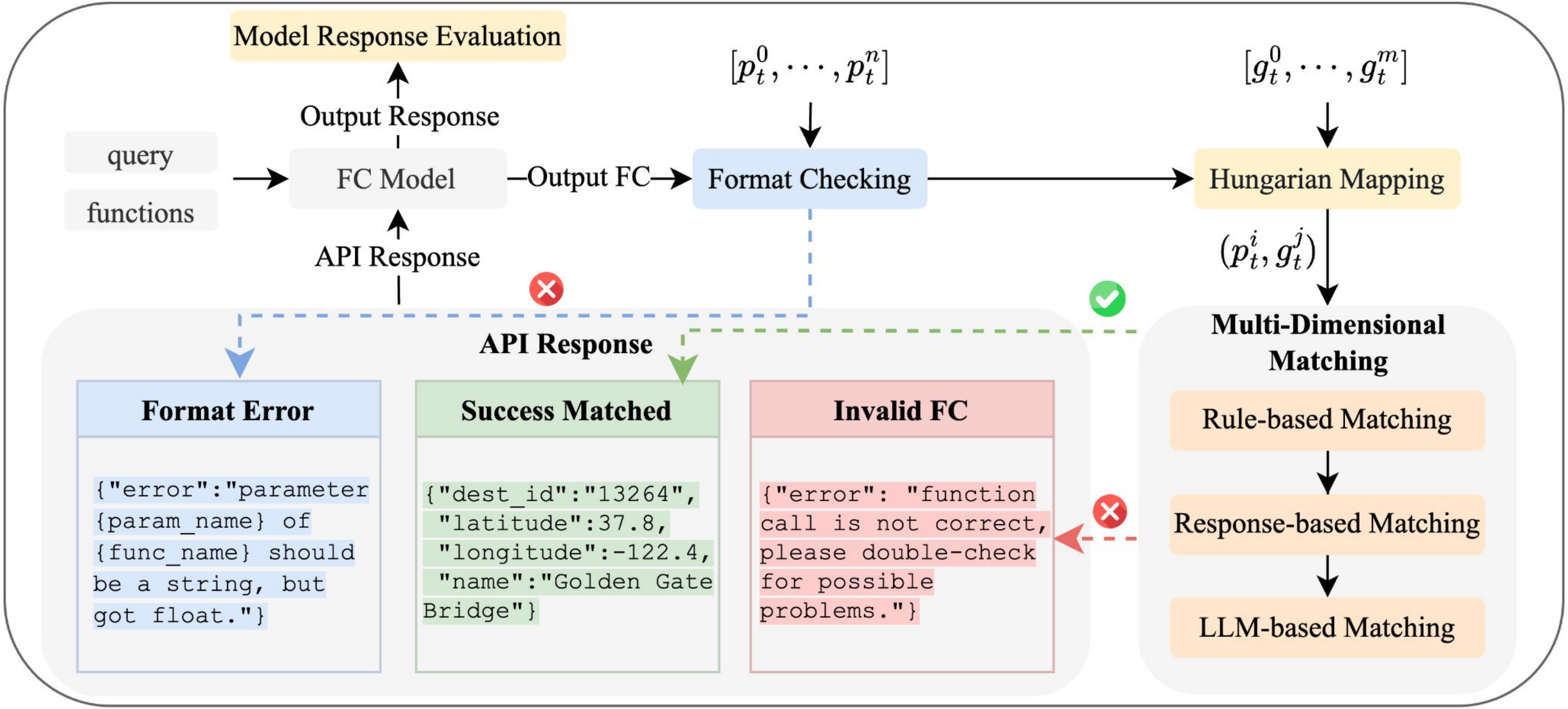


Figure 4: pass^k (–) and pass@k (..) in τ-retail.

ComplexEval (from ComplexFuncBench)



```
{"error":"parameter
{param_name} of
{func_name} should
be a string, but
got float."}
```

```
{"dest_id":"13264",
 "latitude":37.8,
 "longitude":-122.4,
 "name":"Golden Gate
Bridge"}
```

```
{"error": "function
call is not correct,
please double-check
for possible
problems."}
```

COMPONENT v/s E2E.

Focus: **Correct functions usage**

Example metrics:

- AST
- Complex string parameters
- PII leak

Pros:

- Low(er) effort
- Focused

Cons:

- Synthetic, may not be representative or true user needs

Unit tests

Focus: **User intent achieved**

Example metics:

- Correct database state
- Correct response to user
- LLM as Judge of entire flow

Pros:

- Realistic (if done correctly)
- Flexible

Cons:

- High effort

Integration tests

Resolutions

- Single-turn
- Single-turn, multi-step
- Multi-turn, or complete conversation

Summary

- We need **COMPONENT**, low level metric
    - **AST** looks like a good choice
- We need high level **E2E** metric
    - **LLM as judge** over the conversation. Including the internal conversation (multi-step). Still needs work
- Do we need a full user simulator? TBD