KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

The Race to Standardize AI Observability

Liudmila Molkova, Google · Trask Stalnaker, Microsoft



New use cases for observability

Still apply

- Performance
- Alerting
- Troubleshooting

New with AI

- **Evals**
- **Cost tracking**
- **Guardrails**
- **Training data**

...and more

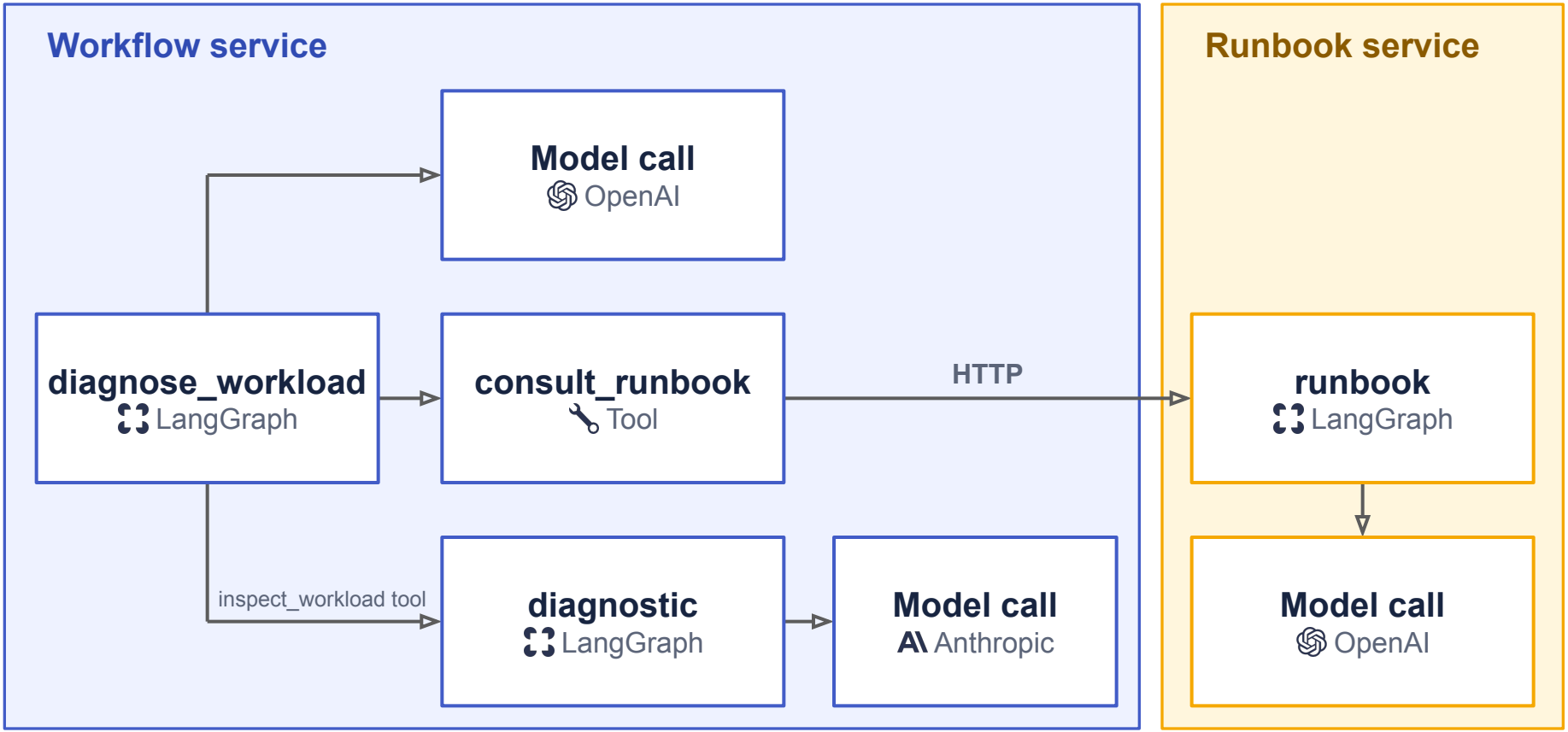
KubeCon CloudNativeCon
North America 2026

Demo

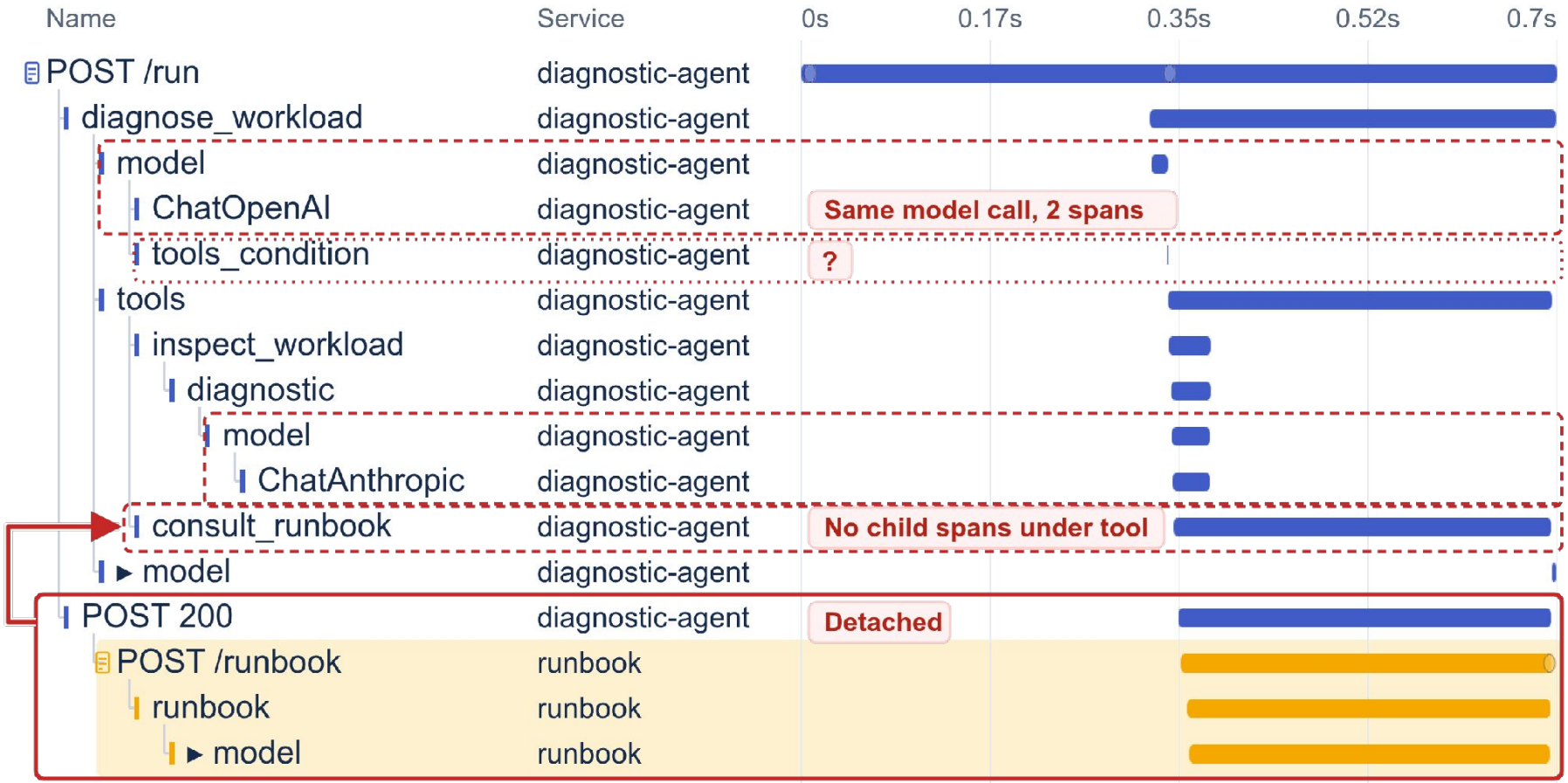


Our application

Investigates Kubernetes incidents: *"Why is the service restarting, and what should I do?"*



Different instrumentations, one broken trace



One standard



Querying token usage

```
SELECT COUNT(*) AS calls,
  SUM("gen_ai.usage.input_tokens") AS input_tokens
FROM span_data
WHERE trace_id = :trace_id
  AND "gen_ai.operation.name" = 'chat';
```

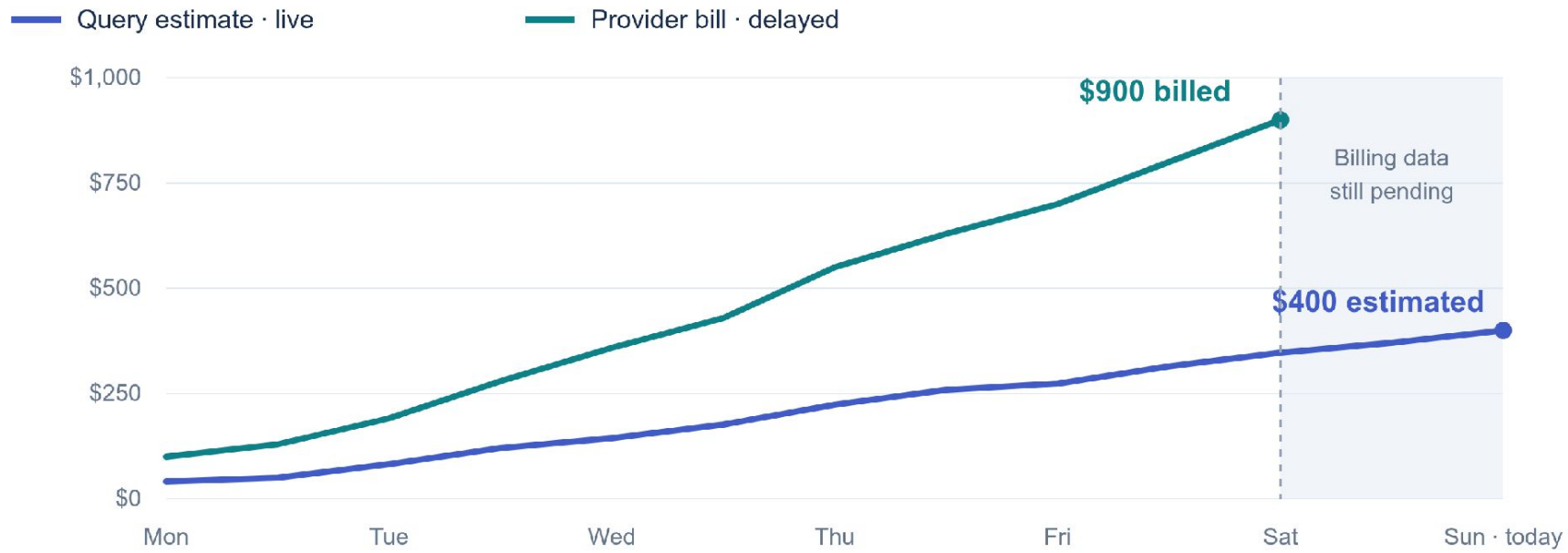
Query result · 1 row

calls	input_tokens
2	280

Dashboard usage vs. reality

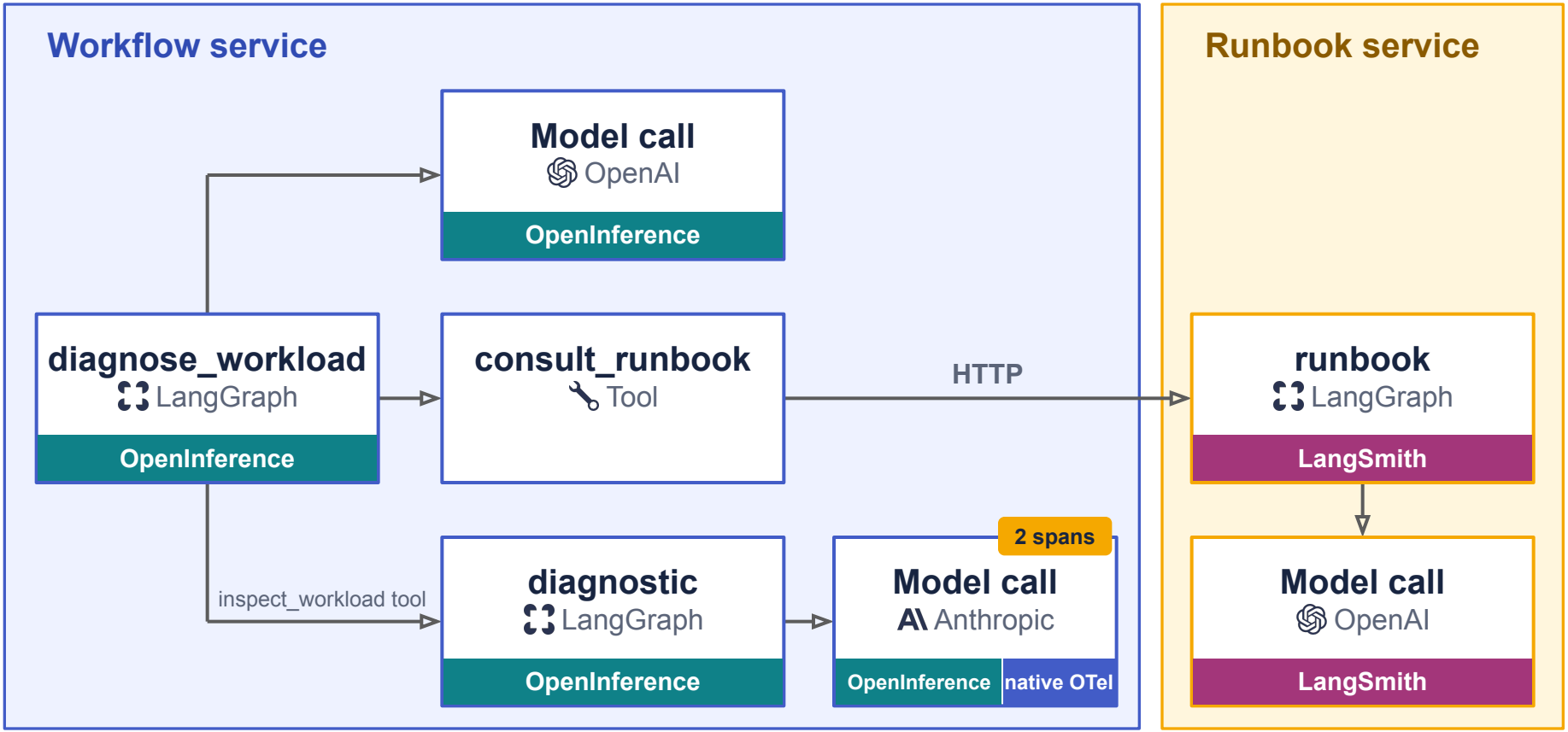
Model spend

Cumulative USD · same application · by usage date



Illustrative data and reporting delay · query usage converted to estimated cost

How did we instrument the application?



Accounting for different schemas

```
WITH calls AS (
  SELECT COALESCE(
    "gen_ai.usage.input_tokens",
    "llm.token_count.prompt"
  ) AS input_tokens
  FROM span_data WHERE trace_id = :trace_id AND (
    "gen_ai.operation.name" = 'chat'
    OR lower("openinference.span.kind") = 'llm'
    OR lower("langsmith.span.kind") = 'llm'
  )
)
SELECT COUNT(*) AS calls, SUM(input_tokens) AS input_tokens
FROM calls;
```

Query result · 1 row

calls	input_tokens
5	900

What did we find?

Standard protocol

✓ All three libraries sent OTLP

Standard structure

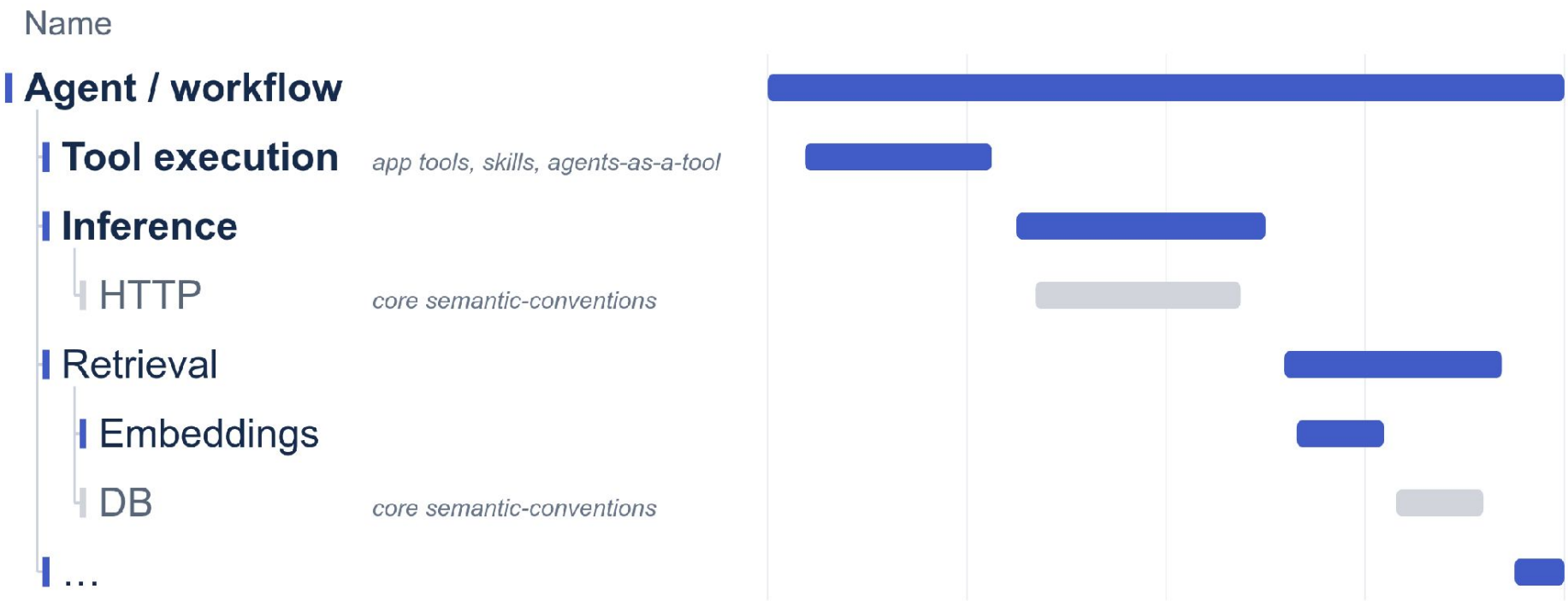
✗ Different span and attribute names

✗ Prompts and responses in different shapes

✗ No AI metrics, so token counts came from spans

GenAI semantic conventions

open-telemetry/semantic-conventions-genai



GenAI attributes

Agent / workflow identity	gen_ai.agent.name	diagnose_workload
Model details and parameters	gen_ai.operation.name	chat
	gen_ai.provider.name	openai
	gen_ai.request.model	gpt-6.1-sol
	gen_ai.response.model	gpt-6.1-sol
	gen_ai.response.id	chatcmpl-4021efaa4f924ee6…
	gen_ai.response.finish_reasons	["tool_calls"]
Token usage	gen_ai.usage.input_tokens	120
	gen_ai.usage.output_tokens	30
Inputs, outputs, and instructions	gen_ai.input.messages	[{"role":"user","parts":[{"content":"Why is checkout restarting, and what should I do?","type":"text"}]}]
	gen_ai.output.messages	[{"role":"assistant","parts":[{"arguments":{"workload":"checkout"},"name":"inspect_workload","type":"tool_call"}]}]
Tool definitions, calls, and results	gen_ai.tool.definitions	[{"name":"inspect_workload","description":"Inspect Kubernetes evidence for a workload."}]

Structured inputs and outputs

gen_ai.input.messages

```
[{"role": "user", "parts": [{
  "type": "text",
  "content": "Why is checkout restarting, and what should I do?"
}]}]
```

gen_ai.output.messages

```
[{"role": "assistant", "finish_reason": "tool_calls", "parts": [
  {"type": "tool_call", "id": "call_diagnose",
   "name": "inspect_workload", "arguments": {"workload": "checkout"}},
  {"type": "tool_call", "id": "call_runbook",
   "name": "consult_runbook", "arguments": {"topic": "OOMKilled"}}
]}]
```

Conversation view

diagnostic-agent: chat gpt-6.1-sol 6483b37 LLM

Input & output | Tools 2 | Details

Duration **1.88ms** · Tokens **360**

Input

User — Message

```
Why is checkout restarting, and what should I do?
```

Assistant — </> Tool calls

```
inspect_workload({"workload": "checkout"})
consult_runbook({"topic": "OOMKilled"})
```

User — Tool response

```
{"evidence": "checkout was OOMKilled; memory
limit is 128Mi and peak usage is 192Mi."}
```

User — Tool response

```
{"recommendation": "Inspect the memory
regression; raise the limit only after
checking cluster capacity."}
```

Output

Assistant — Output

```
Checkout exceeded its 128Mi memory limit. Investigate the regression and review capacity
before changing limits.
```

Standard metrics

Input / output tokens

```
sum(increase({"gen_ai.client.inference.usage.input_tokens"}[5m]))
sum(increase({"gen_ai.client.inference.usage.output_tokens"}[5m]))
```

Cached input tokens

```
sum(increase(
  {"gen_ai.client.inference.usage.cache_read.input_tokens"}[5m]))
```

p95 time to first chunk

```
histogram_quantile(0.95, sum(rate(
  {"gen_ai.client.inference.time_to_first_chunk"}[5m])))
```

p95 duration by operation

```
histogram_quantile(0.95,
  sum by ("gen_ai.operation.name") (rate(
    {"gen_ai.client.inference.duration"}[5m])))
```

Failed calls per second

```
sum(histogram_count(rate(
  {"gen_ai.client.inference.duration", "error.type"!=""}[5m])))
```

KubeCon CloudNativeCon
North America 2026

High-quality instrumentation



How we build instrumentations

open-telemetry/opentelemetry-python-genai

- **One shared GenAI Util** owns spans, metrics, content capture, and context
- **Thin adapters** only map each library's data to GenAI Util types
- **Conformance checks + unit tests** verify each package against the conventions
- **AI-assisted development:** AGENTS.md, skills, and automated pre-review

Inside an instrumentation

```python
def traced_create(wrapped, instance, args, kwargs):
    # Starts the span; on exit ends it and
    # records metrics and errors
    with handler.inference(
        provider="anthropic", request_model=kwargs["model"]
    ) as invocation:

        # Map Anthropic request to GenAI types
        set_request_attributes(invocation, kwargs)

        result = wrapped(*args, **kwargs)

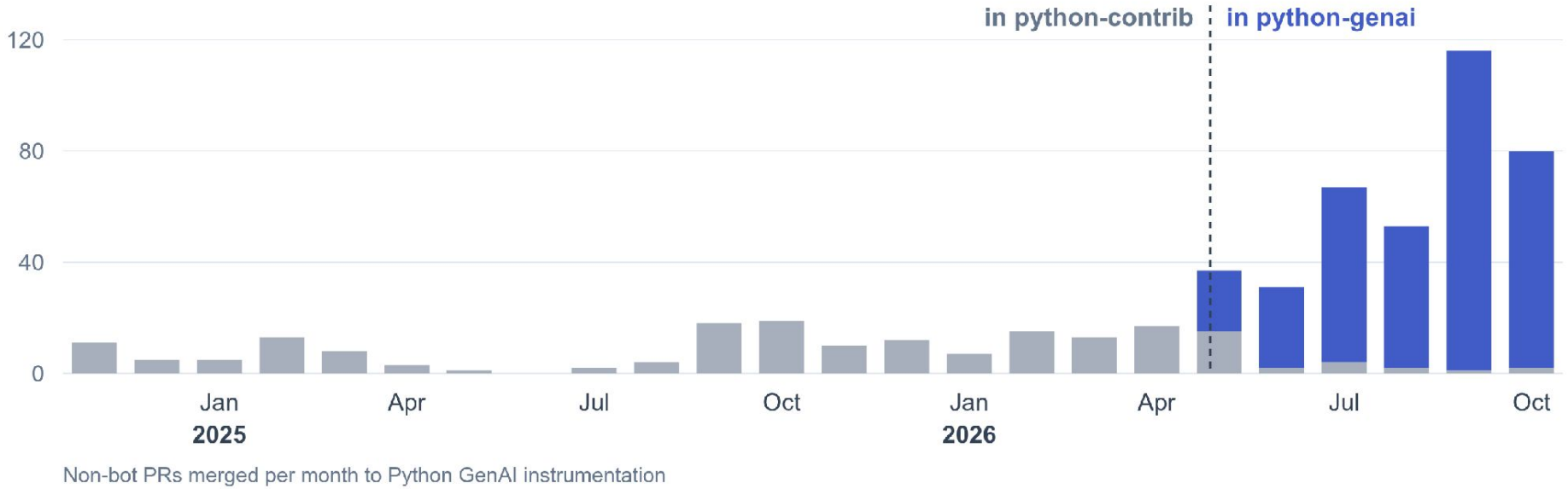
        # Map response: model, usage, finish reasons
        set_response_attributes(invocation, result)
        return result
```

GenAI Util handles

- Spans and attributes
- Metrics
- Content capture
- Errors
- Context

More contributors, faster delivery

- At least 143 contributors to OTel AI work in 2026, 65 new to OTel
- 13 Python instrumentations released in 2026



Non-bot PRs merged per month to Python GenAI instrumentation

Native instrumentation

Libraries that emit OTel themselves are better for users and the ecosystem

Anthropic SDK
Model provider

MCP SDKs
Tool protocol

Microsoft Agent Framework
Agent framework

AWS Strands
Agent framework

Google ADK
Agent framework

Vercel AI SDK
App SDK

KubeCon CloudNativeCon North America 2026

Driving adoption



OpenTelemetry Weaver

- Validates the definitions
- Generates docs & code
- Checks real telemetry

```yaml
spans:
  - type: gen_ai.client.inference
    kind: client
    brief: Client call to a GenAI model
    attributes:
      - key: gen_ai.request.model
        type: string
        requirement_level:
          conditionally_required: If available.
        brief: The GenAI model name.
        examples: 'gpt-4'
      - key: gen_ai.usage.input_tokens
        type: int
        requirement_level: recommended
        brief: Total input (prompt) tokens.
        note: Includes cached tokens ...
```



github.com/open-telemetry/weaver

Conformance testing with Weaver

1. Run a library scenario

```python
from openai import OpenAI

OpenAI().chat.completions.create(
    model="gpt-6-luna",
    messages=[{
        "role": "user",
        "content": "Say hello",
    }],
)
```

2. Check emitted telemetry

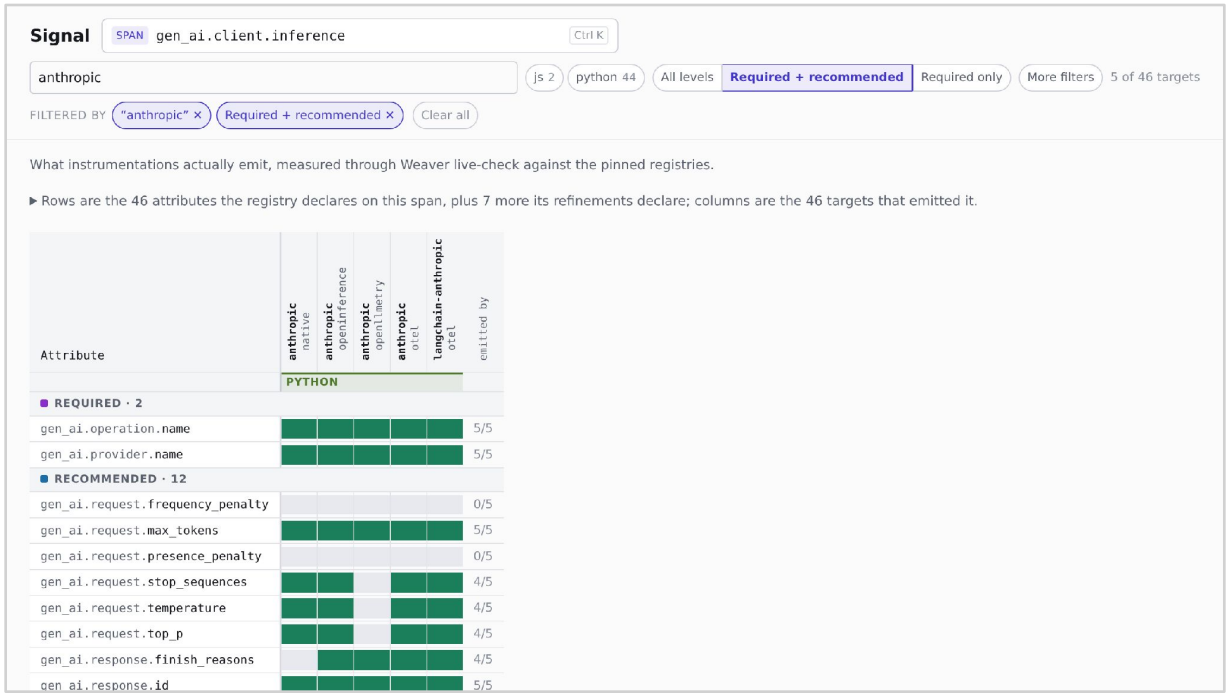
```
weaver registry live-check \
  --registry ./registry \
  --input-source otlp \
  --format json \
  --no-stream \
  --output ./report
```

3. View report

```
{
  "spans": {
    "gen_ai.client.inference": [
      "gen_ai.operation.name",
      "gen_ai.request.model",
      "gen_ai.usage.input_tokens",
      "gen_ai.usage.output_tokens"
    ]
  },
  "findings": []
}
```

github.com/open-telemetry/semantic-conventions-conformance

Conformance Report



Signal SPAN gen_ai.client.inference Ctrl K

anthropic

js 2 | python 44 | All levels | Required + recommended | Required only | More filters | 5 of 46 targets

FILTERED BY "anthropic" × | Required + recommended × | Clear all

What instrumentations actually emit, measured through Weaver live-check against the pinned registries.

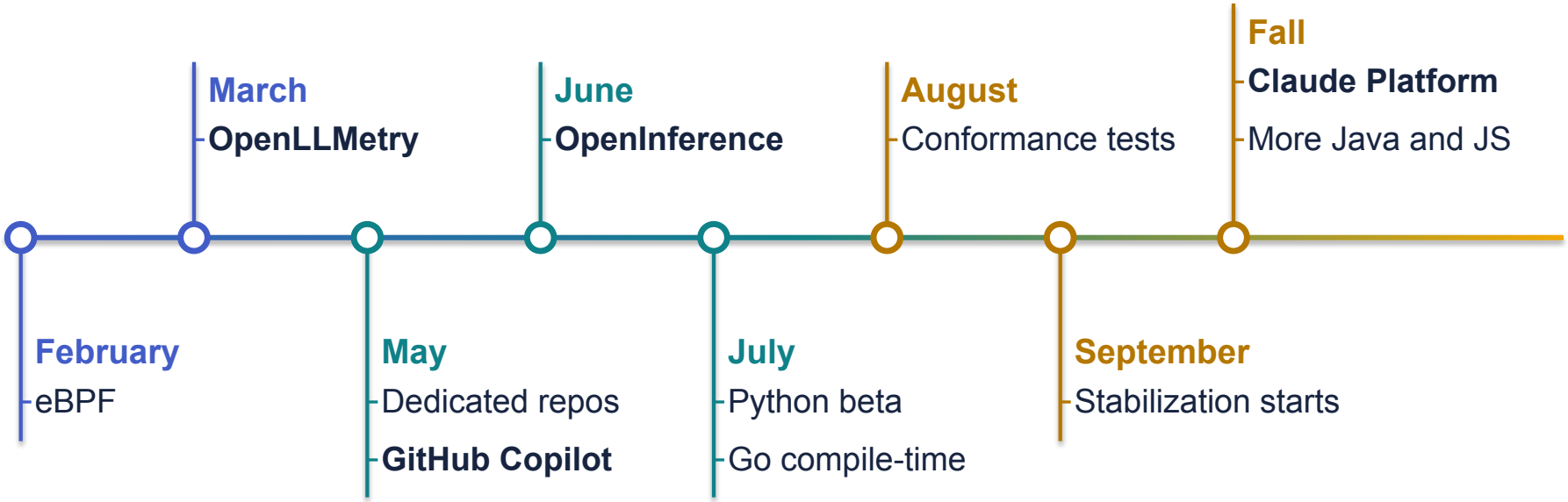
▶ Rows are the 46 attributes the registry declares on this span, plus 7 more its refinements declare; columns are the 46 targets that emitted it.

Attribute	anthropic native	anthropic openinference	anthropic openllmetry	anthropic otel	langchain-anthropic otel	emitted by
	PYTHON					
REQUIRED · 2						
gen_ai.operation.name	■	■	■	■	■	5/5
gen_ai.provider.name	■	■	■	■	■	5/5
RECOMMENDED · 12						
gen_ai.request.frequency_penalty						0/5
gen_ai.request.max_tokens	■	■	■	■	■	5/5
gen_ai.request.presence_penalty						0/5
gen_ai.request.stop_sequences	■	■		■	■	4/5
gen_ai.request.temperature	■	■		■	■	4/5
gen_ai.request.top_p	■	■		■	■	4/5
gen_ai.response.finish_reasons		■	■	■	■	4/5
gen_ai.response.id	■	■	■	■	■	5/5

Explore it yourself

open-telemetry.github.io/semantic-conventions-conformance

GenAI observability milestones in 2026



What's next

End of 2026

Release candidate

KubeCon EU 2027

Stable inference, agents, and tools

NEXT FEATURES TO STABILIZE

- Retrievals & embeddings
- MCP
- A2A
- Live / realtime
- Evaluations
- Skills

STILL TAKING SHAPE

- AAIF collaboration
- Agentic identity & security
- Guardrails
- Harnesses
- HITL
- Costs
- Trajectories

Let's build it together

Use it

Use instrumentations that follow OpenTelemetry GenAI conventions, or emit the conventions natively from your library

opentelemetry-python-genai also Java, JS, Go, Ruby, OBI

Check it

Run the conformance tests against your library or framework

semantic-conventions-conformance

Shape it

Come to the SIG meetings, review the conventions, and bring your use cases

community#calendar

semantic-conventions-genai

#otel-genai-instrumentation

KubeCon CloudNativeCon North America 2026

Thank you!



The Race to Standardize AI Observability