

# Parlant: LLM Agents Built for Control

Designed for Real-World Use. Deployed in Minutes.

---

Finally, LLM agents that actually follow instructions.

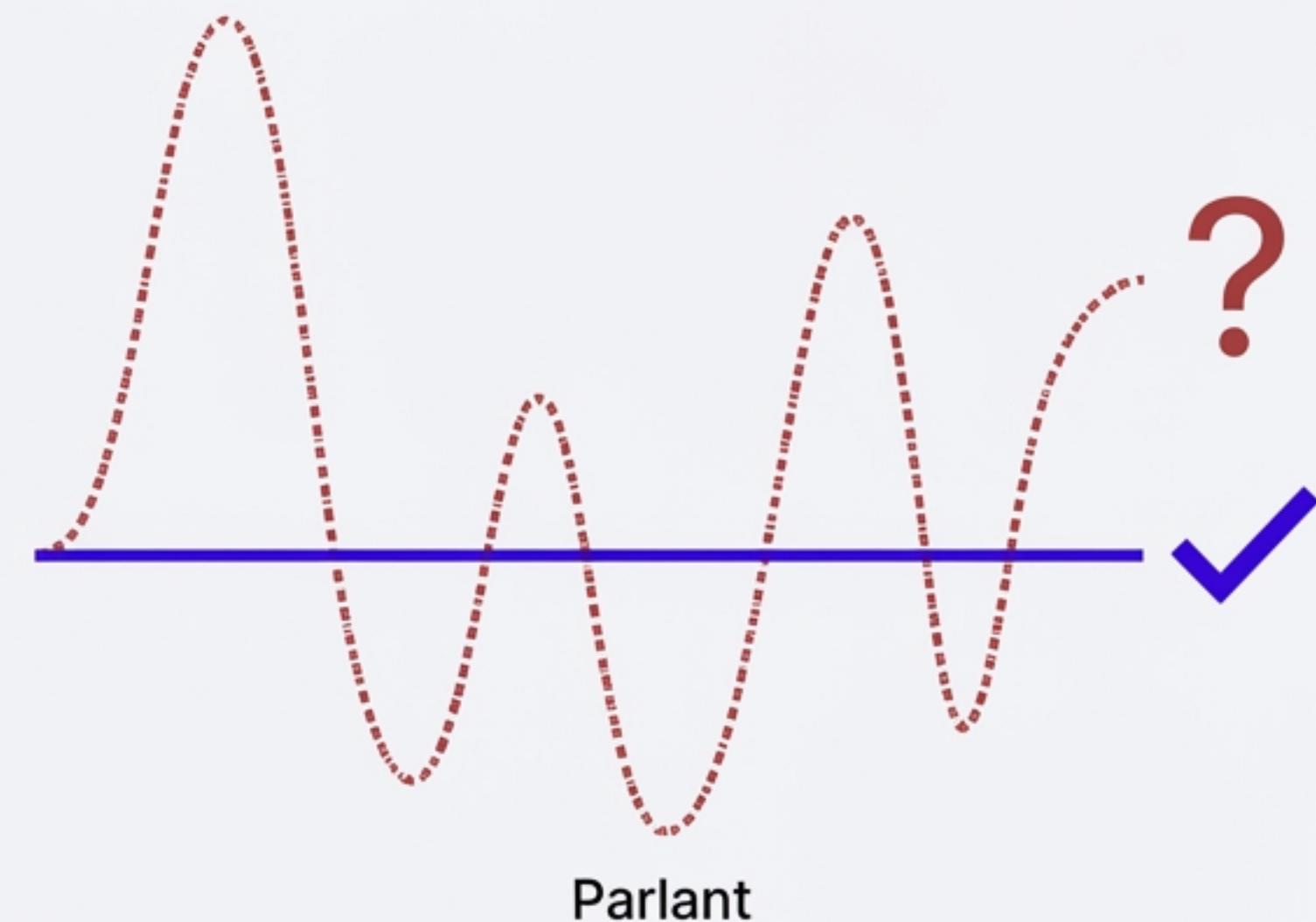
# The Universal Problem: The Stability Gap

You build an agent. It works in testing.  
Then real users touch it.

- It ignores carefully crafted system prompts.
- It hallucinates responses in critical moments.
- It fails to handle edge cases consistently.

**This is the #1 pain point for  
developers building  
production AI agents.**

Traditional Prompting



# Stop Fighting Prompts. Teach Principles.

## Traditional Approach: Cross Your Fingers

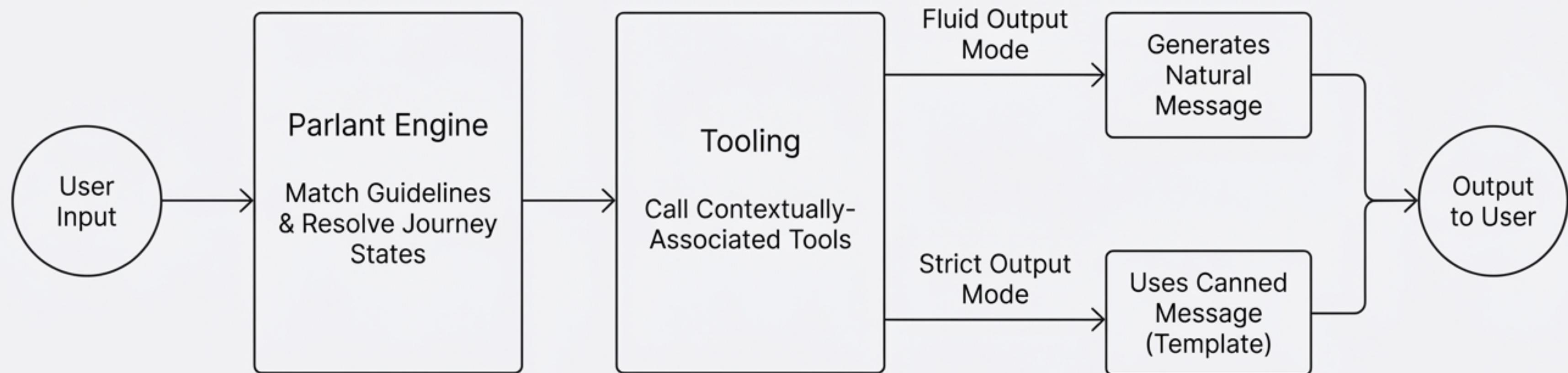
```
system_prompt = "You are a  
helpful assistant. Please  
follow these 47 rules.  
If the user asks X, do Y,  
but only if Z...  
(200 lines omitted)..."
```

## Parlant Approach: Ensured Compliance

```
await agent.create_guideline(  
    condition="Customer asks  
    about refunds",  
    action="Check order status  
    first to see if eligible",  
    tools=[check_order_status],  
)
```

Move from probabilistic prompting to deterministic control.

# Under the Hood: The Parlant Engine



Guidelines and tools are matched and enforced \* before\* generation, keeping agents focused even in complex configurations.

# Core Architecture: Behavioral Guidelines

Easily craft agent behavior. Parlant matches relevant elements contextually to the conversation state. Define rules in natural language, but achieve ensured rule compliance.

```
await agent.create_guideline(  
    condition="User asks about weather", ———— Trigger  
    action="Get current weather and  
        provide tips and suggestions", ———— Instruction  
    tools=[get_weather] ———— Capability  
)
```

Trigger  
Instruction  
Capability

# Managing Conversational Journeys



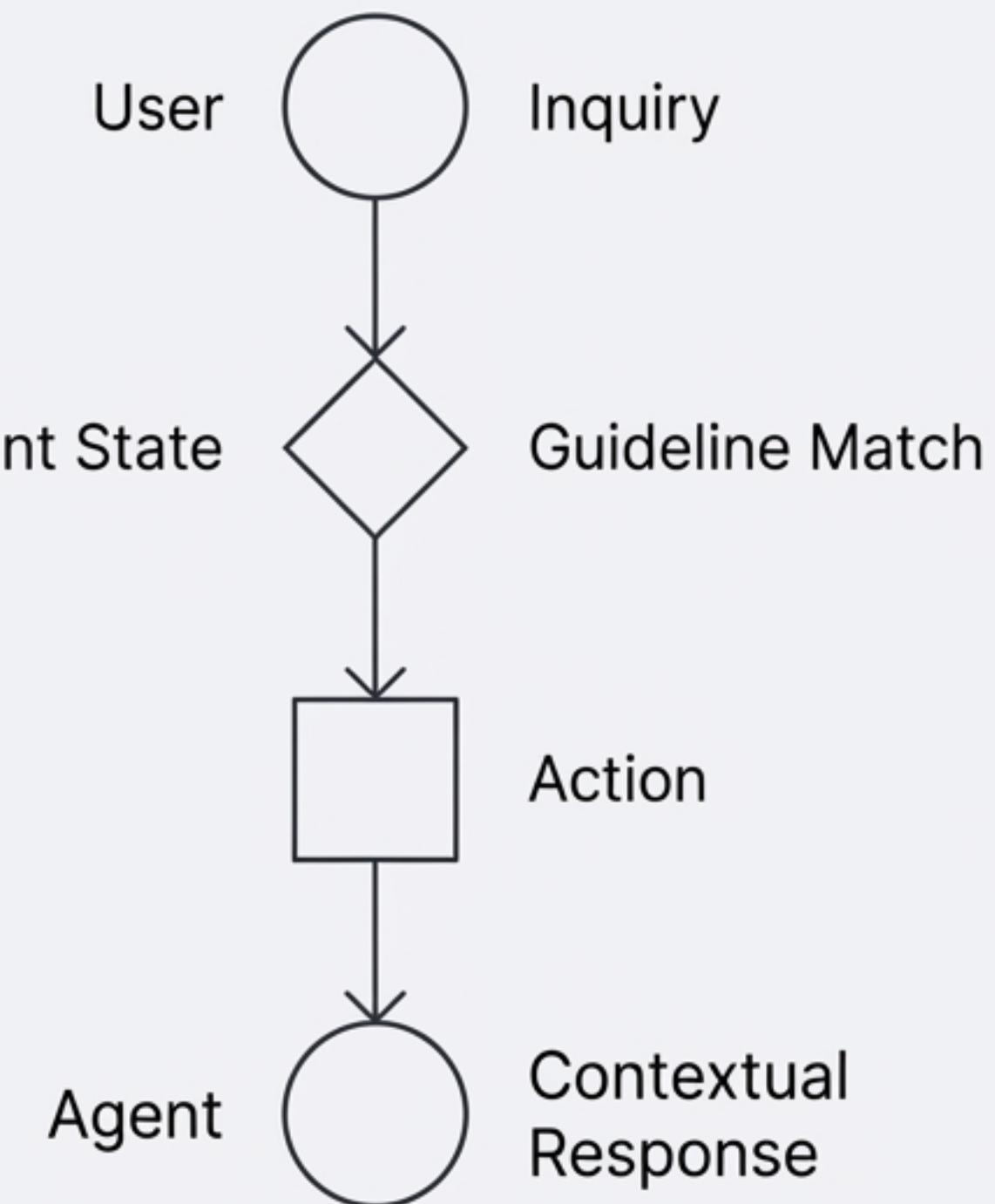
**Journeys:** Define clear customer paths and how the agent responds at each step.



**State Management:** The agent's context is updated on every response.



**Explainability:** Understand exactly why and when each guideline was matched and followed.



# Seamless Tool Integration & Domain Adaptation

```
@p.tool  
async def get_weather(context: p.ToolContext, city:  
    str) -> p.ToolResult:  
    # Your weather API logic here  
    return p.ToolResult(f"Sunny, 72°F in {city}")
```



- Attach external APIs, databases, or backend services.
- **Canned Responses:** Use templates to eliminate hallucinations when strict compliance is required.
- **Domain Adaptation:** Teach the agent domain-specific terminology.

# You Can't Improve What You Don't Measure

## Integrated Testing & Evaluation Framework (LLM-as-a-Judge)

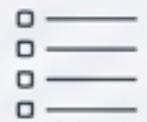
```
# Build conversation history
history = InteractionBuilder().step(CustomerMessage("Man it's cold"))...

# Assert on agent response using LLM-as-a-Judge
await response.should("provide weather details for San Francisco")
```

```
> parlant-test your_tests.py
```



# Why Developers Are Switching

Traditional Frameworks	Parlant
 Write complex system prompts	 Define rules in natural language
 Debug unpredictable behaviors	 Ensured rule compliance
 Scale by prompt engineering	 Scale by adding guidelines
 Cross fingers for reliability	 Predictable, consistent behavior

Production-ready from day one.

# Built for High-Compliance Sectors



## Financial Services

Compliance-first design,  
risk management.



## Healthcare

HIPAA-ready agents,  
patient data protection.



## Legal Tech

Precise legal guidance,  
document review.



## E-commerce

Order processing  
automation, customer  
service at scale.

# Trusted by the Enterprise

“By far the most elegant conversational AI framework that I’ve come across! Developing with Parlant is pure joy.”

Vishal Ahuja, Senior Lead, Customer-Facing Conversational AI @ JPMorgan Chase

**10,000+**  
**Developers**

Inter

**17.7k**  
**GitHub Stars**

Inter

**1.5k**  
**Forks**

Inter

# Get Started in 60 Seconds

```
$ pip install parlant
```

## Try it

```
import parlant.sdk
```

## Test

5-minute quickstart  
available

## Join

Discord Community &  
GitHub Discussions

**Ready to build AI agents that actually work?**