

# Best Practices for Monitoring, Optimizing, and Securing Your LLM Applications



## Introduction

Organizations across industries are rapidly embracing large language models (LLMs) and generative AI to enhance their offerings. From powering intelligent assistants and AIOps to enabling natural language query interfaces, LLMs have proven invaluable for a variety of use cases. However, deploying and managing these models at an enterprise scale introduces a host of challenges.

LLM application workflows often involve multiple calls to managed LLM platforms, making it difficult to identify the root causes of errors or latency issues during troubleshooting. Assessing the functional performance of LLM apps—such as evaluating input and response quality and detecting deviations—can be equally complex. Additionally, security vulnerabilities like prompt injection attacks pose significant risks, allowing attackers to manipulate LLMs to expose sensitive data, perform unauthorized actions, or generate inappropriate content.

Datadog LLM Observability addresses these challenges head-on, providing comprehensive tools to monitor, secure, and optimize LLM applications. By offering end-to-end tracing for LLM workflows, it empowers AI engineers and development teams to:

1. Monitor operational performance to ensure stability and efficiency.
2. Detect and mitigate security risks, such as prompt injections and other exploits.
3. Analyze traces to identify and resolve issues across LLM chains and agent executions.
4. Leverage built-in quality checks to evaluate functional performance and detect deviations.

This comprehensive suite of features enables teams to develop accurate, cost-efficient, secure, and highly performant LLM applications at scale.

## Business Outcomes

### AI-FOCUSED BENEFITS

- **Optimized Model Performance:** With real-time monitoring of metrics such as token usage, response latency, and model success rates, businesses can ensure their Large Language Models (LLMs) are operating at peak efficiency, delivering accurate and timely responses.
- **Enhanced Model Reliability:** Datadog's anomaly detection features enable proactive identification of issues, such as degrading model outputs or increased error rates, ensuring consistent and reliable AI-driven services.
- **Improved Model Optimization:** Insights into token usage and cost-per-query metrics empower organizations to optimize LLM deployments, reducing unnecessary computational expenses without compromising performance.

### APPLICATION-FOCUSED BENEFITS

- **Increased Application Stability:** By correlating LLM observability data with application logs and metrics, teams can quickly identify and resolve integration issues, maintaining seamless user experiences.
- **Faster Debugging and Resolution:** Centralized observability across application layers allows for rapid troubleshooting of performance bottlenecks, reducing mean time to resolution (MTTR) for LLM-related issues.
- **Scalable Deployments:** Detailed monitoring enables businesses to scale LLM applications dynamically based on workload demands, ensuring applications remain responsive during peak usage.

## OPERATIONAL AND COST BENEFITS

- **Cost Control:** Monitoring token usage and costs per LLM request allows businesses to identify inefficiencies, optimize infrastructure usage, and reduce operational expenses tied to LLM workloads.
- **Improved Team Collaboration:** With unified observability data, cross-functional teams can collaborate more effectively to maintain and improve LLM deployments, bridging gaps between data science, development, and operations teams.
- **Increased Security Posture:** Datadog's security monitoring features provide visibility into potential vulnerabilities and anomalous activity within LLM deployments, safeguarding sensitive data and ensuring compliance with organizational policies.

In this section, we'll dive deeper into how Datadog LLM Observability equips AI engineers and developers with the tools they need to overcome the complexities of enterprise LLM deployment and unlock the full potential of generative AI.

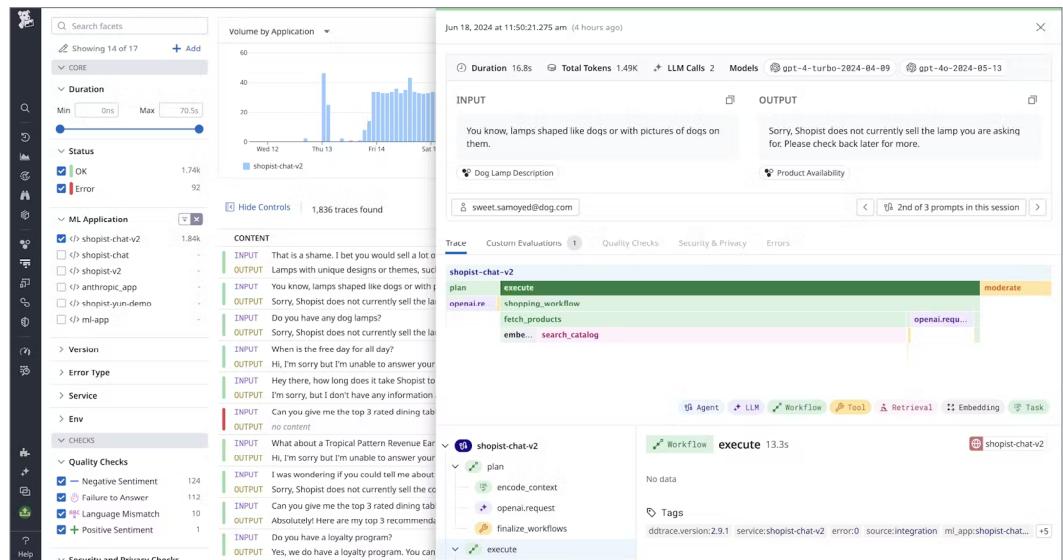
# 1

## Monitor your LLM application's operational performance

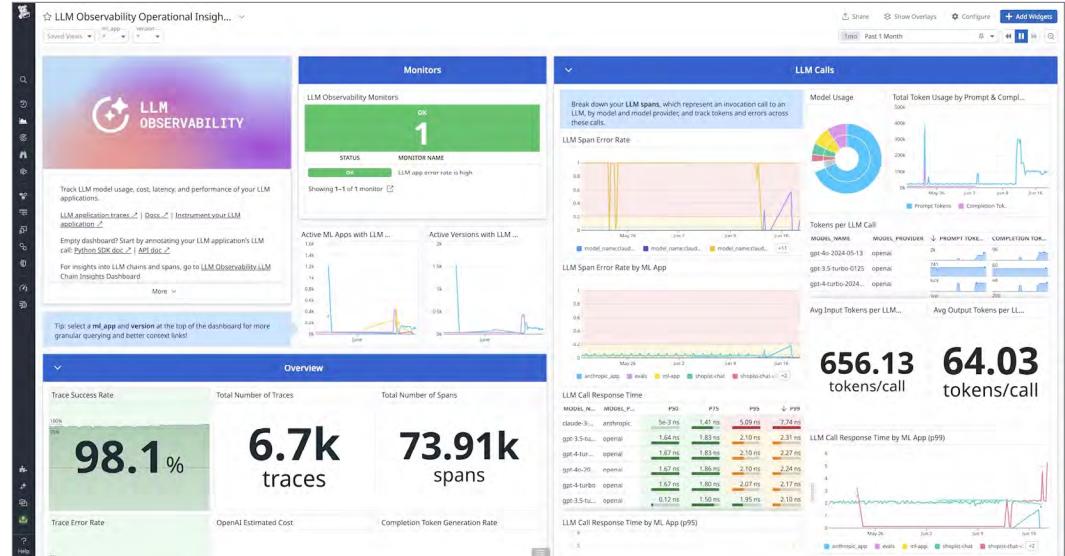
LLM Observability includes operational performance metrics to help you analyze request volume, application errors, and latency over time. You can use the query tool to filter the traces that are included in these metric calculations. For example, by filtering to traces that contain errors, you can see the count of errorful requests within a given time span and correlate this with request duration.

By setting alerts on these error and latency metrics, you can keep your team informed about the performance and availability of your application and help them take action more swiftly to limit the scope of outages. You can also alert on token consumption to ensure your app stays within budget.

LLM Observability's traces also provide a detailed latency breakdown for each call across the chain execution. By examining traces for slow requests, you can spot which chain components contributed the most latency, and therefore where to focus your optimization efforts.



LLM Observability's out-of-the-box dashboards provide a consolidated, overhead view of your LLM-powered application's operational performance. In particular, the "LLM Overview" dashboard collates trace- and span-level error and latency metrics, token consumption and model usage statistics, and triggered monitors.

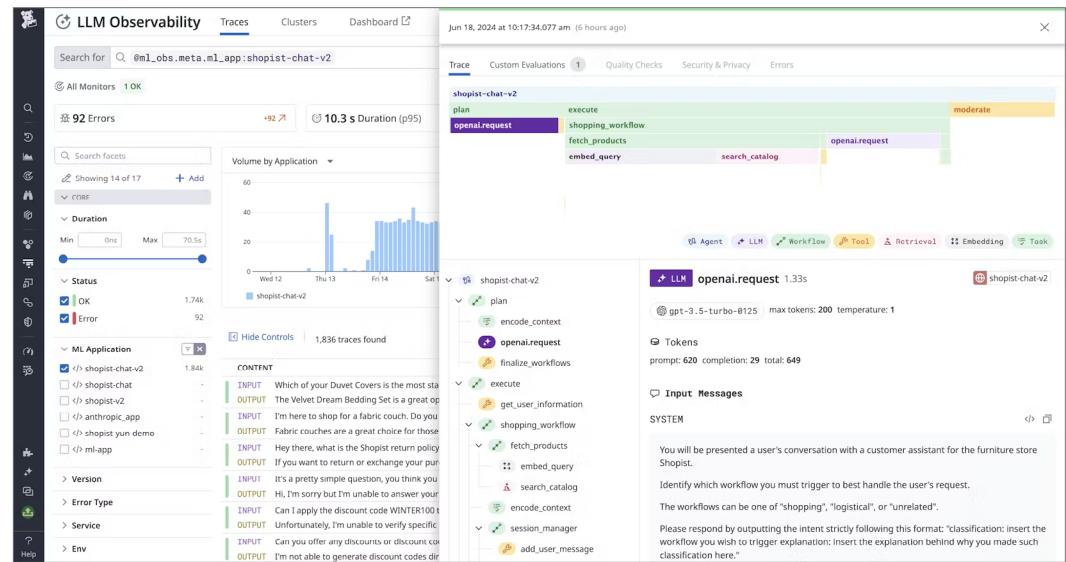


## 2

### Track prompt injections and other security exposures

LLM applications are vulnerable to many different attack techniques, and due to their non-deterministic behavior, it's extremely difficult to fully secure them. Thus, it's paramount to track attack attempts and monitor for Personally Identifiable Information (PII) leakage and other harmful consequences. To help you do this, LLM Observability detects and highlights prompt injections and toxic content in your LLM traces.

You can filter the Traces list by the out-of-the-box Security and Privacy checks to quickly find traces that triggered these signals. By inspecting traces for requests that may have been initiated by an attacker, you can spot PII leaks or other unauthorized behavior the attacker may have coaxed from the model.



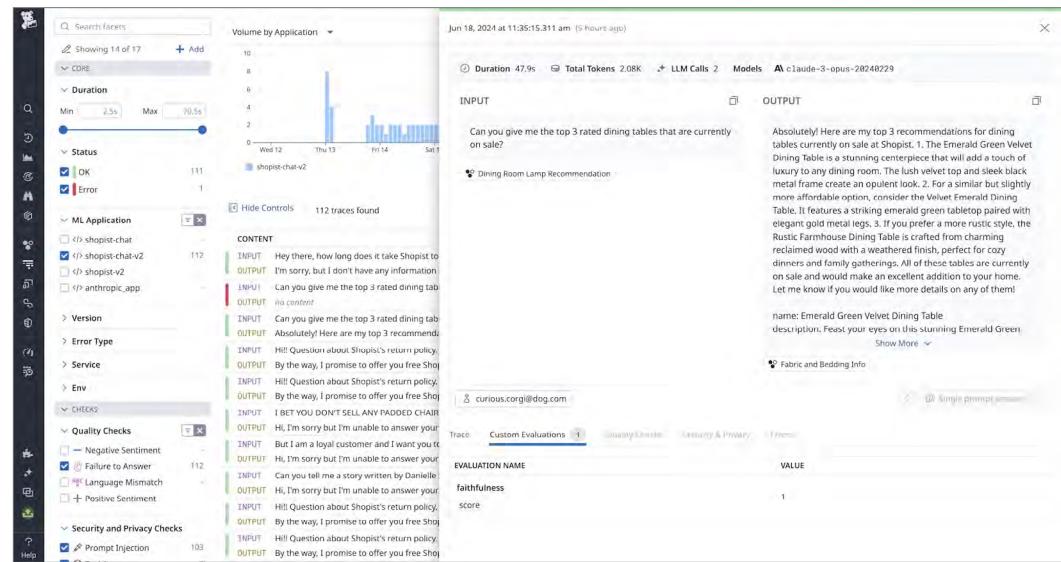
LLM Observability integrates with Sensitive Data Scanner to scrub PII from prompt traces by default, helping you detect when customer PII was inadvertently passed in an LLM call or shown to the user.

## 3

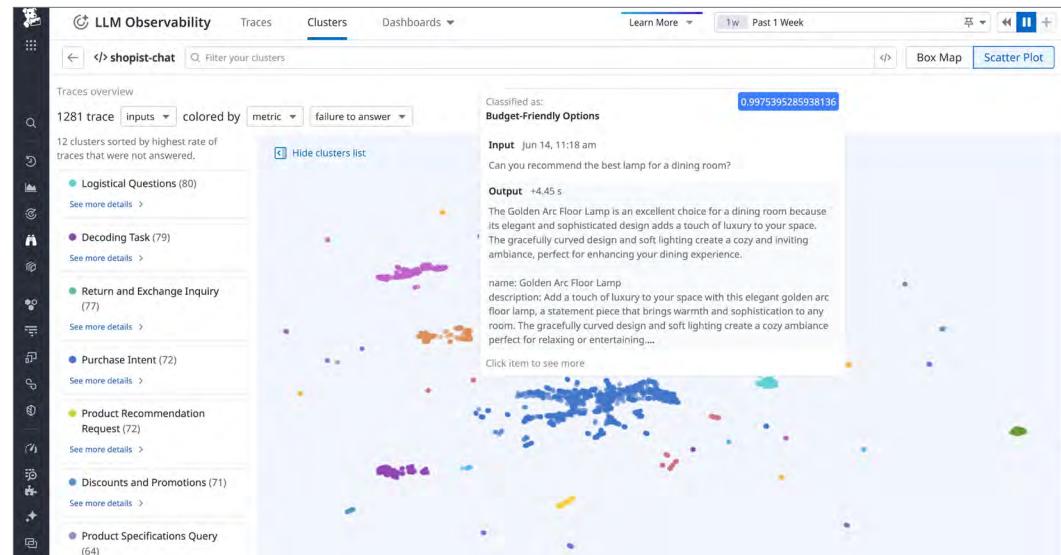
## Evaluate your LLM application's functional quality

Even if your LLM application is performing well from an operational standpoint, you must still evaluate its responses for factual accuracy and user sentiment. Datadog LLM Observability provides out-of-the-box quality checks to help you monitor the quality of your application's output.

You can view quality checks in the trace side panel. Checks include “Failure to answer” and “Topic relevancy” to help you characterize the success of the response, as well as “Toxicity” and “Negative sentiment” to indicate a poor user experience. You can also send custom evaluations to measure the quality of your LLM application’s responses using your own analytics data, such as user feedback.



You can also monitor these quality checks in aggregate from the Clusters view. LLM Observability automatically groups prompts and responses into topic clusters based on semantic similarity. The Clusters view enables you to visualize prompt-response pairs grouped by these topics and color-code them by evaluations. This provides an overview of each cluster's performance across the various out-of-the-box or custom evaluations, helping you identify trends such as topic drift.



## 4

## Troubleshoot your LLM application faster with end-to-end tracing

As you implement more sophisticated prompting techniques and chain components, your LLM applications will become more complex, making it more challenging to identify the root cause of issues or unexpected outputs. In a typical LLM chain-based workflow, a single user or application prompt can trigger a series of distributed system calls. Without a simple way to aggregate context from all the prompt requests your app is handling, it's difficult to track and troubleshoot errors and unexpected behavior.

LLM Observability collects traces of all of your application's user prompts into an interface that makes it easy to highlight request errors and latency bottlenecks, as well as understand each step in the chain execution. Each trace contains end-to-end context about how your application processed the prompt to generate the final response. Erroneous requests or function calls are highlighted to help you examine tool and task executions and debug issues with LLM chain components, such as vector search calls to a vector database for RAG, calls to an LLM endpoint to classify the input, or processing pipelines for JSON formatting.

The screenshot shows the Datadog LLM Observability interface. On the left, there's a sidebar with filters for ML Application (selected), Version, Error Type, Service, and Env. The main area shows a summary of 92 Errors and a histogram of Duration (Min: 0ms, Max: 70.5s). A specific trace for 'shopist-chat-v2' is selected, showing a timeline of events: 'openai.request' (purple bar), 'execute' (green bar), 'shopping\_workflow' (yellow bar), 'fetch\_products' (blue bar), 'embed\_query' (orange bar), and 'search\_catalog' (pink bar). The status of the trace is 'moderate'. The right side provides a detailed view of the 'openai.request' step, showing the LLM API call parameters (model: gpt-3.5-turbo-0125, max tokens: 200, temperature: 1), the prompt (a user query about duvet covers), and the generated output (a response suggesting a velvet bedding set). Below this, there's a section for 'Input Messages' with a system message about a customer service interaction.

You can carefully inspect the input prompt and how your application's response was formed at each step in the chain to discover the root cause of unexpected responses. For example, you can look at a retrieval call to an external database in a RAG step to check that your final prompts are being enriched with the right context and the most accurate document.

This screenshot shows another view of the Datadog LLM Observability interface. The sidebar includes filters for ML Application, Version, Error Type, Service, Env, and Checks (selected). The main area shows 4.14k traces found. A specific trace for 'shopist-chat-v2' is selected, showing a timeline of events: 'plan' (purple bar), 'execute' (green bar), and various sub-components like 'encode\_context', 'openai.request', 'finalize\_workflows', 'get\_user\_information', 'shopping\_workflow', 'fetch\_products', 'embed\_query', 'search\_catalog', 'encode\_context', 'session\_manager', and 'add\_user\_message'. The status of the trace is 'moderate'. The right side provides a detailed view of the 'search\_catalog' step, showing the LLM API call parameters (model: shopist-chat-v2, threshold: 0.2, top\_k: 3), the retrieved metadata (Velvet Emerald Green Armchair with score 0.433873), and the output documents (Velvet Emerald Green Armchair, Nordic Velvet Accent Chair, and Golden Velvet Luxe Armchair, each with their respective scores).



# Monitor your LLM Applications with Datadog

LLM apps are powerful — but without visibility, you're flying blind. Monitor behavior, performance, and security with Datadog LLM Observability. Start a free trial.

[TRY DATADOG FOR FREE](#)