Gartner®

Introduce AI Observability to Supervise Generative AI

Published 5 September 2023 - ID G00790301 - 38 min read

By Analyst(s): Carlton Sapp

Initiatives: Analytics and Artificial Intelligence for Technical Professionals; Drive Quantifiable Value With D&A Solutions for the Business

Generative AI technologies have vast potential, but their lack of transparency and explainability fuels security concerns and distrust. Technical professional AI engineers and architects should use this analysis to present AI observability on GenAI solutions as a basis for supervising AI behavior.

Overview

Key Findings

- Generative AI (GenAI) can produce unintended outcomes like not following user instructions, hallucinations, or copyright infringement. Technical architectural elements are available that allow for the ability to observe and direct the execution of GenAI activities to avoid unintended outcomes.

- AI observability is an emerging approach that enables oversight and supervision of GenAI systems. It allows technical and business stakeholders to gain better understanding and control of the generated output.

- Using AI observability helps CDAOs and technical teams scale up GenAI securely and responsibly. It helps to achieve greater AI adoption, trust, security and responsible AI.

- A key technique to instrumenting GenAI for AI observability uses a modular open system approach (MOSA) to design observable architectural components that not only deliver key insights, but nudge GenAI models enough to influence outcomes.

Recommendations

- Reduce the risks and enhance the trust of your generative AI systems by building observability into the solution architecture of AI-powered solutions. Use the AI observability reference architecture outlined in this research to better understand generated output and highlight potential issues.

- Instrument your generative AI solutions for observability by introducing a good enterprise model performance management software that can curate an inventory of observability metrics. Extend its capability to include preventative controls to deter or prevent unintended events.

- Provide a human-led oversight or trustee service to the practice of AI observability and assign independent verification and validation resources to all interfaces and components of the AI observability architecture framework.

- Communicate shared semantics and guidelines from the AI observability repository that serve as resources for developers and end users.

Analysis

C-suite leaders believe that AI will be the technology most significantly impacting their industry over the next few years (see 2023 CEO Survey — The Pause and Pivot Year). The AI technology showing the most promise in impacting industries across the world is generative AI (GenAI). In April 2023, a study from the National Bureau of Economic Research concluded that GenAI-based conversational assistance improved productivity by 14% on average (see Generative AI at Work). The study also showed that GenAI-assistance improved customer sentiment and improved employee retention.

Generative AI refers to AI techniques that learn a representation of artifacts from data, and use it to generate brand-new, completely original artifacts that preserve a likeness to original data. These artifacts can serve benign or nefarious purposes. Generative AI can produce totally novel media content (including text, image, video and audio), synthetic data and models of physical objects (Hype Cycle for Artificial Intelligence, 2023).

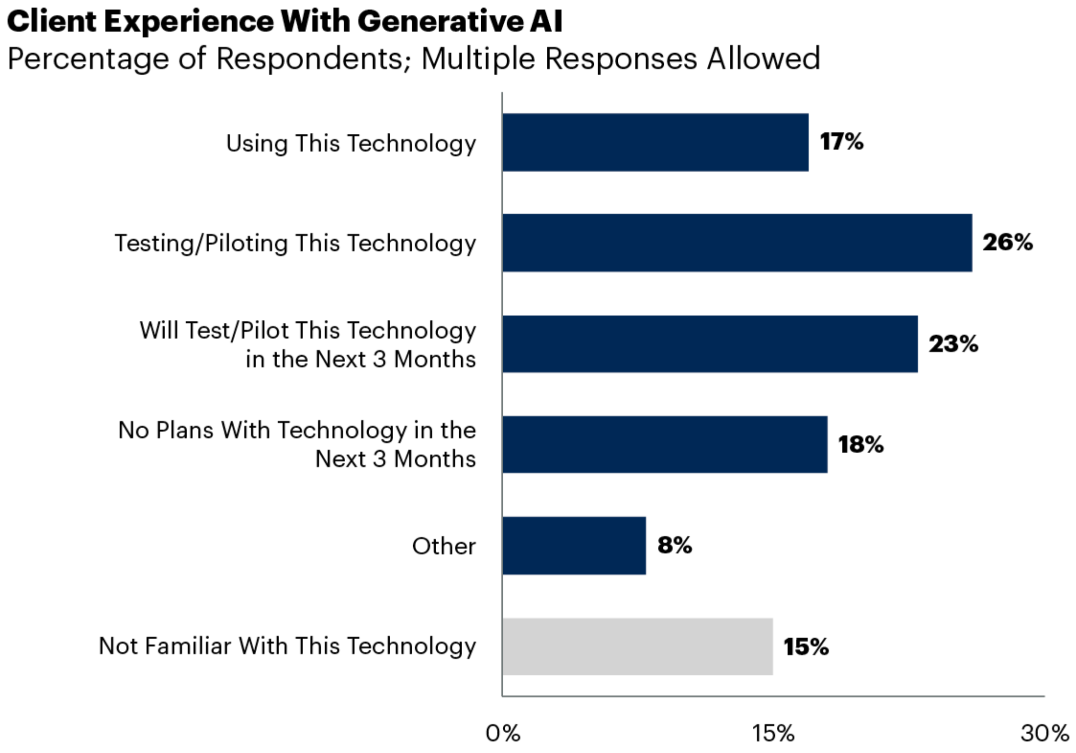
Since generative AI creates original artifacts based on enormous and often unknown data sources, we must give extra attention to its risks and how we implement frameworks to address them.

The Rise of Generative AI and Business Risks

The rise of generative AI has been accompanied by an increase in the prevalence of business risks and ethical issues, like the generation of false content or violation of data privacy (see Human Programmers Are Behind AI's Behavior and Misbehavior, Albuquerque Journal). Governments are racing to develop priorities and standards for AI. For example, The White House Office of Science and Technology Policy recently released to private industries a Request for Information for techniques to mitigate the risk of AI. Additionally, private sector organizations are searching for ethics advisory councils, containing a mix of internal and external experts, to weigh in on AI development (see Managing the Risks of Generative AI, Harvard Business Review).

This causes many technology and business leaders to tread lightly with their adoption of the technology. Many Gartner clients in IT leader roles and business leader roles said they are still in a "testing and piloting" phase in a recent survey, as highlighted in Figure 1. Testing and piloting GenAI is a safe way to collect information about the AI's expected behavior in a controlled environment.

Figure 1: Client Experience With Generative AI



n = 695, Gartner clients in ITLeader Roles and Business Leader Roles with recent engagement with Gartner content, excludes “unsure”

Q: Which of the following best describes your experience with generativeAI (such as OpenAI ChatGPT, Google Bard, AWS CodeWhisperer, etc.)?

Source: 2023 Gartner LRP Voice of Client Content

Note: Approximately 5% selected more than one response.

790301_C

Gartner

Part of the need to test and pilot these technologies comes from the challenges of observing and directing GenAI execution. Organizations continue to struggle with governing these types of AI systems because of limited knowledge and transparency of generative AI’s deep machine learning models’ internal workings. It is also difficult to prevent unintended content outcomes. A lack of visibility into these systems can lead to unintended outcomes, such as the AI:

- **Making up facts** because the models and their objectives are not aligned to the users, or validating truth with an unknown source.
- **Not following instructions** because the models are not properly trained to do so.

- **Hallucinating from data** because of model overfitting or when new data isn't captured in the training set or there is limited visibility into tracking the change in very large sets of input content (see Artificial Hallucinations in ChatGPT: Implications in Scientific Writing, Cureus)
- **Hallucinating from models** because of bias or how the model is trained.
- **Producing synthetic distortion** when generating media because of limited validation and verification into generated output.
- **Becoming wicked AI**, also known humorously as the "elephant in Cairo" (see Pachydermic Personnel Prediction, Byte), where there is no safe stopping point for the generative AI solution, resulting in unbounded distributions of content.

Technical professionals must enable greater visibility into these systems. Organizations that enable observing and directing the execution of their generative AI solutions are more likely to move from technology testing and piloting to use.

It is not always necessary to know the internal workings of generative AI models to manage their behavior. Gartner clients are seeing success in instrumenting their generative AI solution architectures with observability and controllability. This instrumentation allows for greater management of GenAI solutions when there is limited understanding of how and why AI models produce certain content.

While there are many practices, solutions and technologies designed to manage the opaqueness of GenAI solutions or explain the results of a model (e.g., responsible AI practices, explainable AI), technical professionals need more comprehensive technical practices in the architecture of AI solutions so that those solutions are "helpful, honest, and harmless" (see Training Language Models to Follow Instructions With Human Feedback, Cornell University).

A challenge with establishing technical practices to ensure that generative AI solutions are helpful, honest, and harmless is the various implementation strategies for integrating generative AI solutions with existing architectures. Different implementation strategies have different impacts on technical architectures. Many generative AI solutions are leveraging commercially available application programming interfaces (APIs) to access generative AI models (e.g., tools from Hugging Face). For example, some Gartner clients leverage the OpenAI API to gain access to GPT-3, GPT-3.5 and GPT-4 in order to accelerate their implementation of generative AI solutions. Other clients simply integrate and customize OpenAI's ChatGPT technologies. Both options have differing architectural considerations that present different challenges.

This research considers the use case of commercially available APIs to access generative pretrained transformer (GPT) models, the use of cloud computing platform technologies like ChatGPT and the development of fit-for-purpose generative AI solutions. However, there are other GenAI/large language model (LLM) platforms and use cases, such as MosaicML (now a part of Databricks), that support LLMs, and they are customizable with your own data. Generative AI platforms are a prime starting point for evaluating GenAI model behavior because of the control you gain for building and training the models against the Databricks Lakehouse Platform.

Why Should Architects Focus More on Generative AI Solutions vs. Traditional AI?

AI architects pay more attention to architecting generative AI solutions over traditional AI for several reasons.

First, the data being generated and the data being used by generative AI deserve greater architectural considerations to support explainability and traceability to referenced data sources. Generative AI models that are trained on internal or external data may need the capability to identify relevant data sources and original content at scale. Since GenAI is generating new content, that content may need to reference its source to invoke trust. Comparatively, traditional AI focuses more on discriminative tasks like describing and deducing from existing data. Modern data architecture will see the most impact since the new data requires additional scrutiny when it becomes source data or integrated into business workflows. Additionally, engineers will need greater and sometimes different investments in data infrastructure and governance to support future use of generative AI (see FlexGen: High-Throughput Generative Inference of Large Language Models With a Single GPU, arXiv).For example, data architects are designing private data puddles to manage internal and generated content from generative AI to tune LLMs and protect against sensitive information leaks. Additionally, data strategies must incorporate practices of data governance on training data used for pretrained models (e.g., LLMs). Data and analytics (D&A) professionals should recognize the symbiotic relationship.

Second, the growing sophistication in the AI models and algorithms representing different architectures (e.g., input, encoder/decoder) that ingest input data slightly differently for performance reasons and requires additional attention. These performance enhancements provide flexibility for architects to design alternative architectures that are more fit for purpose.

For example, architecting for machine learning regression models used in sales forecasting requires different architectural considerations than for AI-generated hyperpersonalized content. Many GenAI LLMs require high-end accelerators to support high computational throughput (see FlexGen: High-Throughput Generative Inference of Large Language Models With a Single GPU, arXiv).

Thirdly, GenAI technologies are introducing more multimodal capabilities, like GPT-4. This means our architectures should describe the interaction between the interconnected models of different data types and AI agent-to-agent interactions to better manage AI behavior.

For example, GPT-4 can accept images and text as input to generate a response. Architects must account for the integration of different data input types when designing generative AI solutions.

All three of these features allow for an opportunity for architects to observe and direct the execution of a task or activity. This is the supervision needed: the ability to observe and direct the execution of generative AI activities. Observability plays a larger role in supervising GenAI technologies. Software companies have been successfully implementing observability in IT architectures to supervise and explain complex system states (see How to Implement Observability in Your IT Architecture, Red Hat and Introducing the Splunk Observability Suite, Splunk). We can extend these frameworks and apply them to GenAI systems.

As previously mentioned, a lack of supervision can lead to unintended behaviors. This prompts technical professional AI architects to continuously ask:

- What practices can I use to better understand how generative AI is behaving so I can take appropriate action if needed?
- How do we implement those practices into the design of our generative AI systems to ensure that observability is ingrained in our solution architectures?

Introducing AI Observability

What Is AI Observability?

> **AI observability is the ability to manage and assess the behavior of an AI solution based on components interfacing with the AI models so that you gain better understanding and control of the output.**

It is especially useful throughout the AI solution life cycle because it can establish the context, set expectations and drive the outcomes of the solution.It provides instrumentation that enables the ability to capture key insights of your AI solution to stakeholders while enabling the ability to individually validate and verify each component as they interact with your AI models.While AI observability works for all AI solutions, it is more imperative in opaque AI systems like generative AI because it does not require extensive knowledge of AI models' internal workings. Instead, AI observability enables the influence through management and controllability of the resources influencing generative AI models.

Those resources evolve into key architecture principles of AI observability, and they are modeled in generative AI systems as a part of the solution architecture when designing and implementing generative AI.

Gartner's AI Observability Principles for Generative AI

AI observability has a set of architecture principles that shape organizations' GenAI strategy and execution. These principles should be deep-seated in generative AI solution architectures because they help guide decision making and the solution's operation. However, these principles can be leveraged for traditional AI where the output of AI is driving business workflows (e.g., health diagnostic tools to analyze medical images).

- **Principle 1:** All data feeding or training models are accessible and monitored.

This may be difficult if using closed models from commercially available generative AI technologies like ChatGPT. Gartner recommends making a vigorous effort to negotiate visibility into data used to train these commercial technologies or introducing data proxies to validate outputs against credible data sources. There are also emerging techniques for open-source models, like model cards and data sheets that explain in detail how the model was designed, trained and implemented (see Datasheets for Datasets, arXiv and Model Cards for Model Reporting, arXiv).

- **Principle 2:** Control methods and procedures provide feedback and are usable to influence model activities.

Control system architectures have long been used for high-level supervision of machines and processes, but their application to GenAI is becoming more popular. For example, in the field of human-to-computer interaction, GenAI principles have been designed to control against potential harms that may arise from a generative model's hazardous output, misuse or potential for human displacement (see Toward General Design Principles for Generative AI Applications, arXiv).

- **Principle 3:** Generative AI models and their life cycles are traceable.

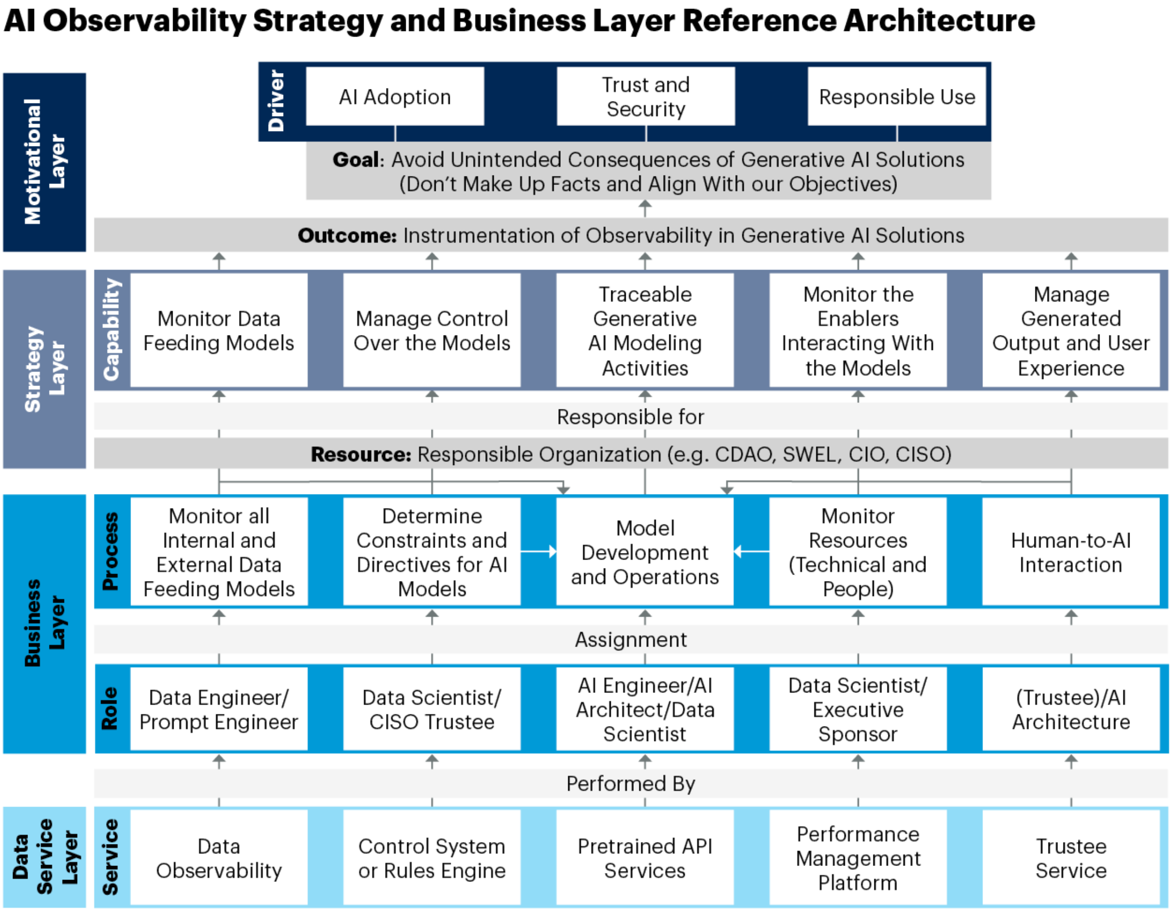
Model traceability includes tracing how various components in a solution interact with generative AI models.

- **Principle 4:** Technology and human enablers interactions are detectable and provide a shared foundation for generative AI model activities.

- **Principle 5:** Generated output is testable and has agent trustees accountable for quality to ensure output is helpful, honest, and harmless.

Agent trustees can be human or AI-based agents. AI-based agent methods like constitutional AI offer self-improvement through the use of AI assistants that engage with harmful queries by explaining their objections to them (see Constitutional AI: Harmlessness from AI Feedback, arXiv).All principles are applicable, regardless of the implementation strategy. They apply when creating and training your own GenAI solutions in-house from scratch, when using a hybrid approach where you adapt an existing solution or when leveraging an existing solution with no changes or customizations.Figure 2 depicts these architecture principles as capabilities in a reference architecture for generative AI, with sample roles and processes typically assigned to them. However, the roles assigned to the process may vary, depending on the organization and the organization structure.The reference architecture illustrates why AI observability is important and depicts its reasons, goals, principles and requirements. The architecture starts with outlining the drivers of AI observability with planned outcomes and capabilities.

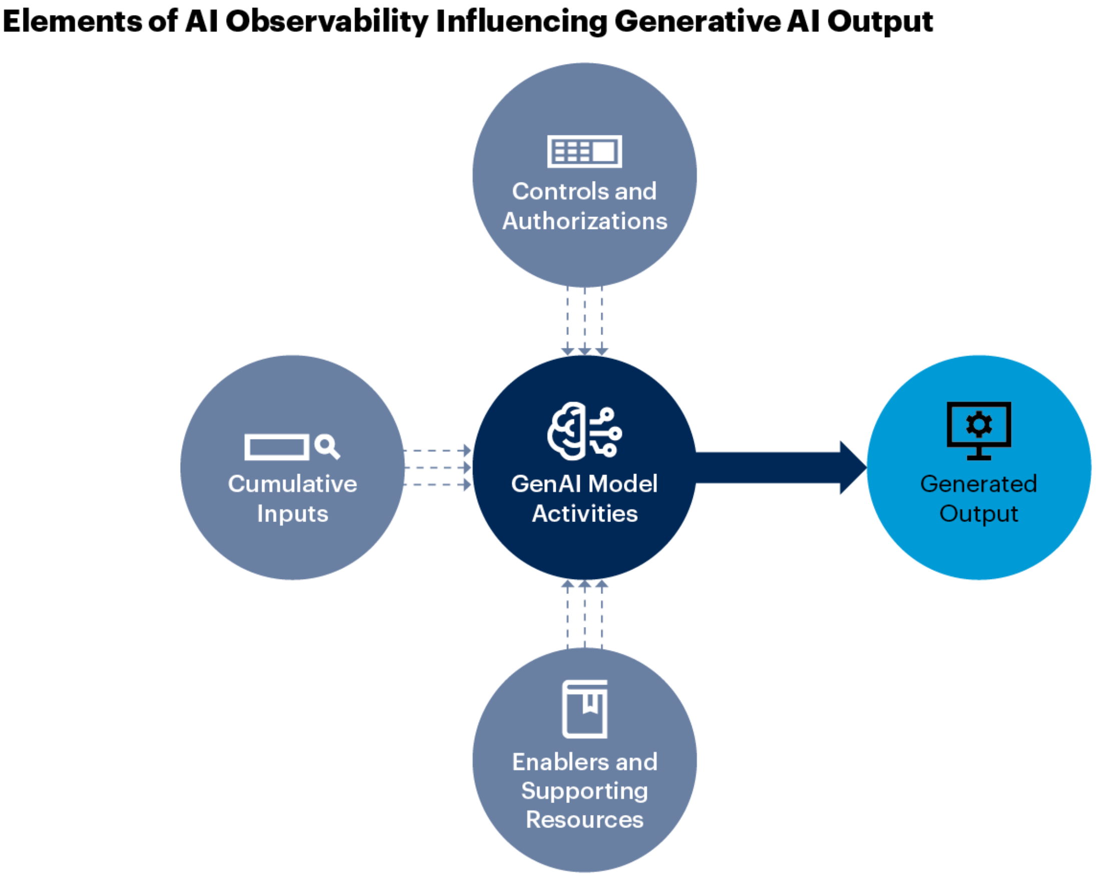
Figure 2: AI Observability Strategy and Business Layer Reference Architecture


AI Observability Strategy and Business Layer Reference Architecture
Motivational Layer
Driver
AI Adoption
Trust and Security
Responsible Use
Goal: Avoid Unintended Consequences of Generative AI Solutions (Don't Make Up Facts and Align With our Objectives)
Outcome: Instrumentation of Observability in Generative AI Solutions
Strategy Layer
Capability
Monitor Data Feeding Models
Manage Control Over the Models
Traceable Generative AI Modeling Activities
Monitor the Enablers Interacting With the Models
Manage Generated Output and User Experience
Responsible for
Resource: Responsible Organization (e.g. CDAO, SWEL, CIO, CISO)
Business Layer
Process
Monitor all Internal and External Data Feeding Models
Determine Constraints and Directives for AI Models
Model Development and Operations
Monitor Resources (Technical and People)
Human-to-AI Interaction
Assignment
Role
Data Engineer/ Prompt Engineer
Data Scientist/ CISO Trustee
AI Engineer/AI Architect/Data Scientist
Data Scientist/ Executive Sponsor
(Trustee)/AI Architecture
Performed By
Data Service Layer
Service
Data Observability
Control System or Rules Engine
Pretrained API Services
Performance Management Platform
Trustee Service

Source: Gartner
790301_C

As described in Figure 2, there are composition relationships between the generative AI model activities and one of its integrated process parts. The purpose of this relationship is to show that generative AI models and the processes involved are tied together. For example, the lifetime of generative AI modeling activities and its internal and external data input processes are tied together because AI models must have data feeding them.In its simplest form, AI observability is a model-centric approach. Information goes into AI models and a generated output is produced. The purpose of architecting for AI observability is to provide sufficient implementation details on elements influencing the output for technical professional roles.A more general view of the elements influencing generated output is shown in Figure 3.

Figure 3: Elements of AI Observability Influencing Generative AI Output



Elements of AI Observability Influencing Generative AI Output
Controls and Authorizations
Cumulative Inputs
GenAI Model Activities
Generated Output
Enablers and Supporting Resources

Source: Gartner
790301_C

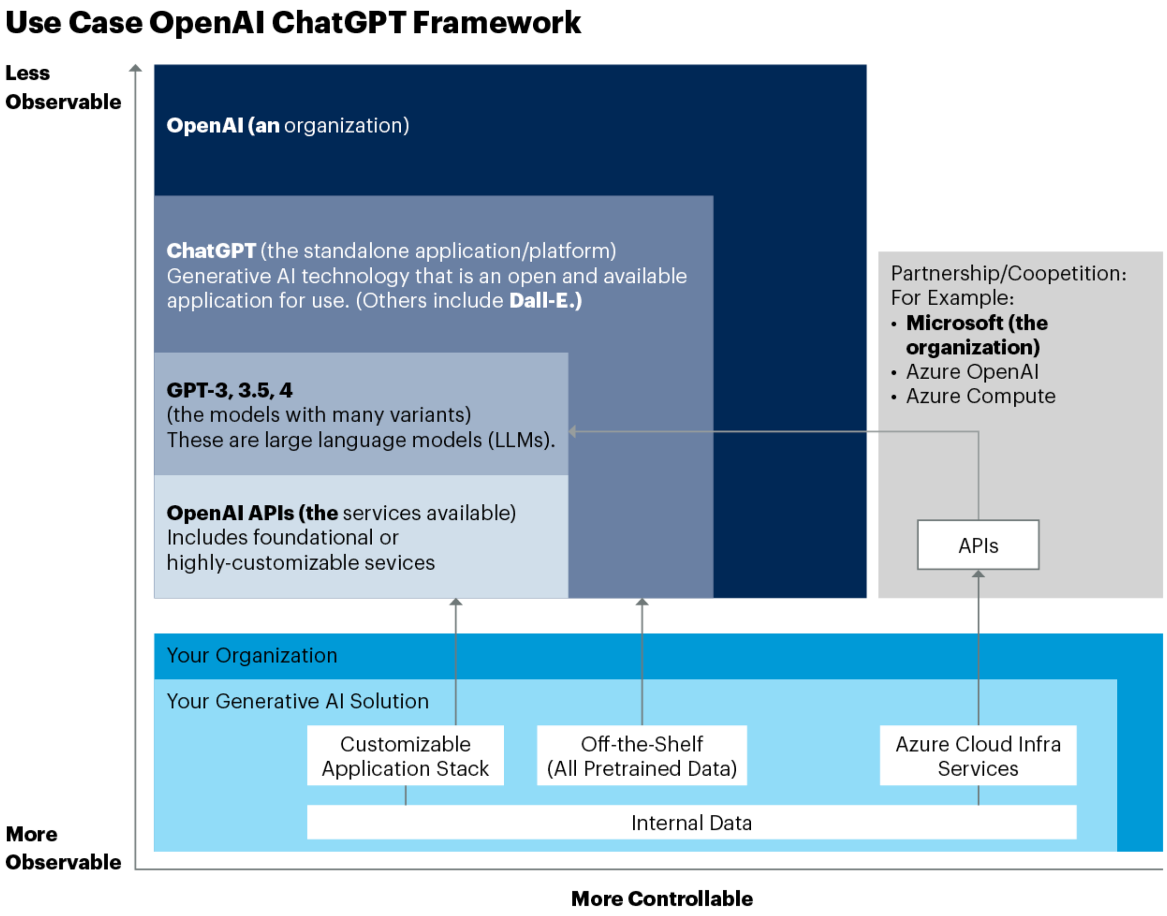
AI observability has four key parts that are described in this research. The remainder of this research will describe the parts in more detail.

- **Layers** to provide an abstraction of generative AI solution features.
- **Components** to identify and capture services that interact with each other.
- **Interfaces** to capture the behavior between generative AI solution components and layers.
- **Repository** to store metrics collected from the interfaces, components and layers.

All four parts of AI observability can be described in the solution architecture for generative AI, and all four parts may be used to support implementation and training of these systems. However, the focus of this research is on the implementation of the solutions. Technical professionals mostly want to observe the implementation because it interacts directly with the end user or business workflows.

To illustrate how to apply AI observability in your generative AI solution, the remainder of this research will focus on one of the most common use cases, OpenAI's ChatGPT, GPT models and LLM algorithms. Figure 4 provides the context and use case for this research, which is OpenAI's ChatGPT and how its components relate to an organization's generative AI solution stack. However, there are several use cases that have different levels of observability and controllability depending on what public or private deep learning services are required to support the GenAI solution.

Figure 4: Use Case Example: OpenAI ChatGPT Framework


Use Case OpenAI ChatGPT Framework
Less Observable
More Observable
More Controllable
OpenAI (an organization)
ChatGPT (the standalone application/platform) Generative AI technology that is an open and available application for use. (Others include Dall-E.)
GPT-3, 3.5, 4 (the models with many variants) These are large language models (LLMs).
OpenAI APIs (the services available) Includes foundational or highly-customizable sevices
Partnership/Coopetition: For Example:
• Microsoft (the organization)
• Azure OpenAI
• Azure Compute
APIs
Your Organization
Your Generative AI Solution
Customizable Application Stack
Off-the-Shelf (All Pretrained Data)
Azure Cloud Infra Services
Internal Data

Source: Gartner
790301_C

In the use case described in Figure 4, the goal of an organization is to interact with a public service GenAI technology, like ChatGPT from OpenAI. The organization plans to use a public GPT model to support a customer service business workflow. Since the models are public from an external organization, the architecture is less observable than an architecture using internal GPT models.

AI Observability Layers

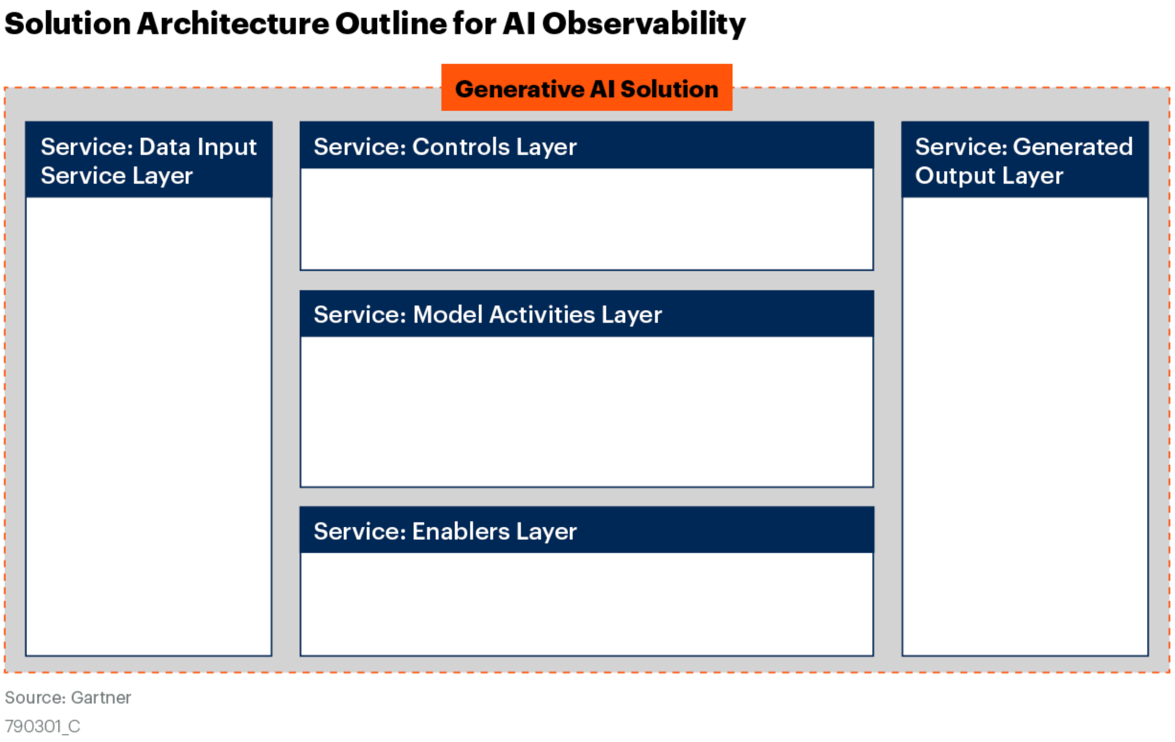
AI observability layers are an abstraction of features that groups components interacting with AI models. They describe the logical characteristics of the solution architecture needed to support AI observability for generative AI solutions. The layers:

- Guide AI developers and data scientists on how to monitor a group of generative AI features.

- Describe the interaction between different features that produce an output or result.

A layer of AI observability enables a set of encapsulated components and features to interoperate. It is used to depict the services interacting with the AI models. For example, generative AI solutions are systems composed of components with many features that interact with each other, like data processing interacting with machine learning model life cycles, feedback loops and application interfaces. Technical professionals must understand and manage the components within layers to better understand their behavior and to depict which services are needed to implement a generative AI solution.Figure 5 outlines the five layers that represent a solution architecture outline framework for AI observability. These layers work together to provide insights about the behavior of your AI system by capturing information that is interacting between components.

Figure 5: Solution Architecture Outline for AI Observability

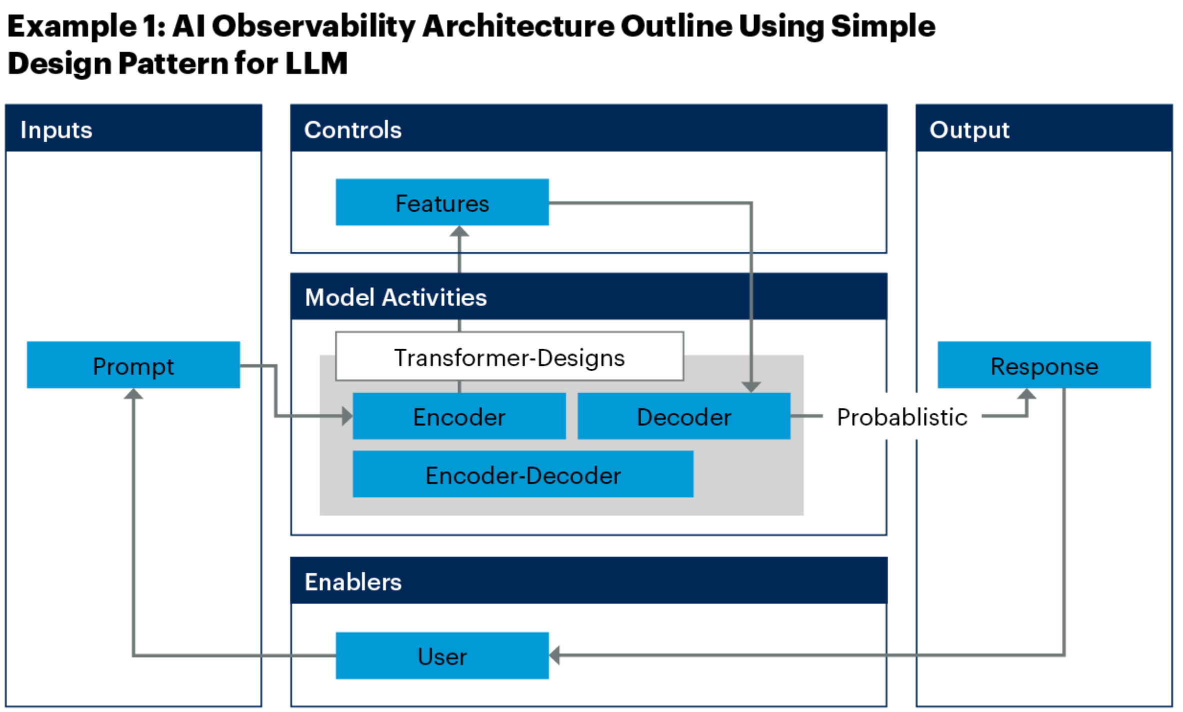

Solution Architecture Outline for AI Observability
Generative AI Solution
Service: Data Input Service Layer
Service: Controls Layer
Service: Model Activities Layer
Service: Enablers Layer
Service: Generated Output Layer
790301_C
Gartner

Source: Gartner

Each layer serves a unique purpose in influencing the behavior of your AI solutions without disrupting the development of operational life cycles. For example, data components in the data input service layer feed the modeling activity tasks, like LLMs using a transformer architecture to process and generate data in sequence for text, in the Model Activities layer. Control and enabling components also feed the modeling activity components. The modeling activities feed the generated output.Figure 5 provides a use case using a simple design pattern for LLMs. It describes how the reference architecture outline described in Figure 4 can be used to support the implementation of ChatGPT using one of Gartner's low-difficulty LLM design patterns (see AI Design Patterns for Large Language Models). Gartner has identified common design patterns that can be used to guide architects with high-level use case scenarios. The design patterns for LLMs are reusable solutions to common AI and software design problems. These LLM design patterns are trisected by difficulty (low, medium, high) and give immediate guidance for organizations to execute on short-term opportunities. They also provide a target state to plan ahead for more complex implementations. Organizations must consider these design patterns as they plan and design their LLM implementations.

The outline in Figure 6 shows how a common design pattern can be deconstructed to observe and direct the execution of various tasks within a transformer architecture. Each component of the design pattern is decoupled into a management framework layer. The focus on the decomposition is in the transformer architecture. Transformer architecture embodies a type of deep neural network that computes a numerical representation of training data (see Innovation Insight for Generative AI). Manipulating the transformer architectures is nontrivial. However, influencing the architectures with design patterns that highlight what and how data interacts with the encoder and decoder is showing promising results for Gartner clients developing their own LLM design strategy.

Figure 6: AI Observability Architecture Outline Using Simple Design Pattern for LLM



Source: Gartner
790301_C

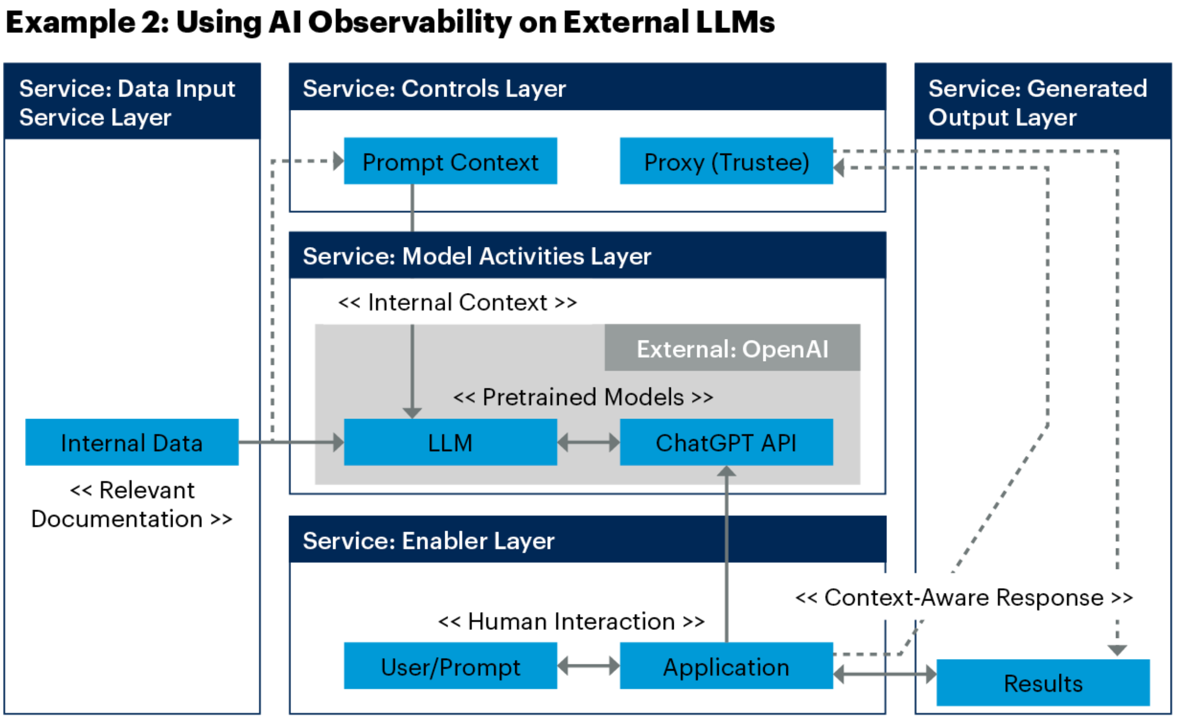
The first step in establishing AI observability within your architectures is to decompose the model architectures used in generative AI systems. The reference architectures outlined in Figures 5 and 6 demonstrate how clients are deconstructing generative AI systems (like ChatGPT using LLMs) into observability layers that provide task visibility and allow humans or agents of systems to influence how AI executes those tasks. In another example, the AI observability outline is used to highlight external components interacting with an application.

- The **data input layer** captures all internal data used to customize or tune an external LLM hosted by OpenAI.
- The **enabler layer** captures the user and application as key enablers to the model and results.
- The **model layer** captures the external model services used to interact with the rest of the generative AI solutions.

- The **control layer** captures controls for the prompt feeding of the LLM. This layer also captures an options proxy service to control what results are presented as outputs before feeding back to the application.

Figure 7 provides another example of the AI observability outline as a framework.

Figure 7: Example 2: Using AI Observability on External LLMs

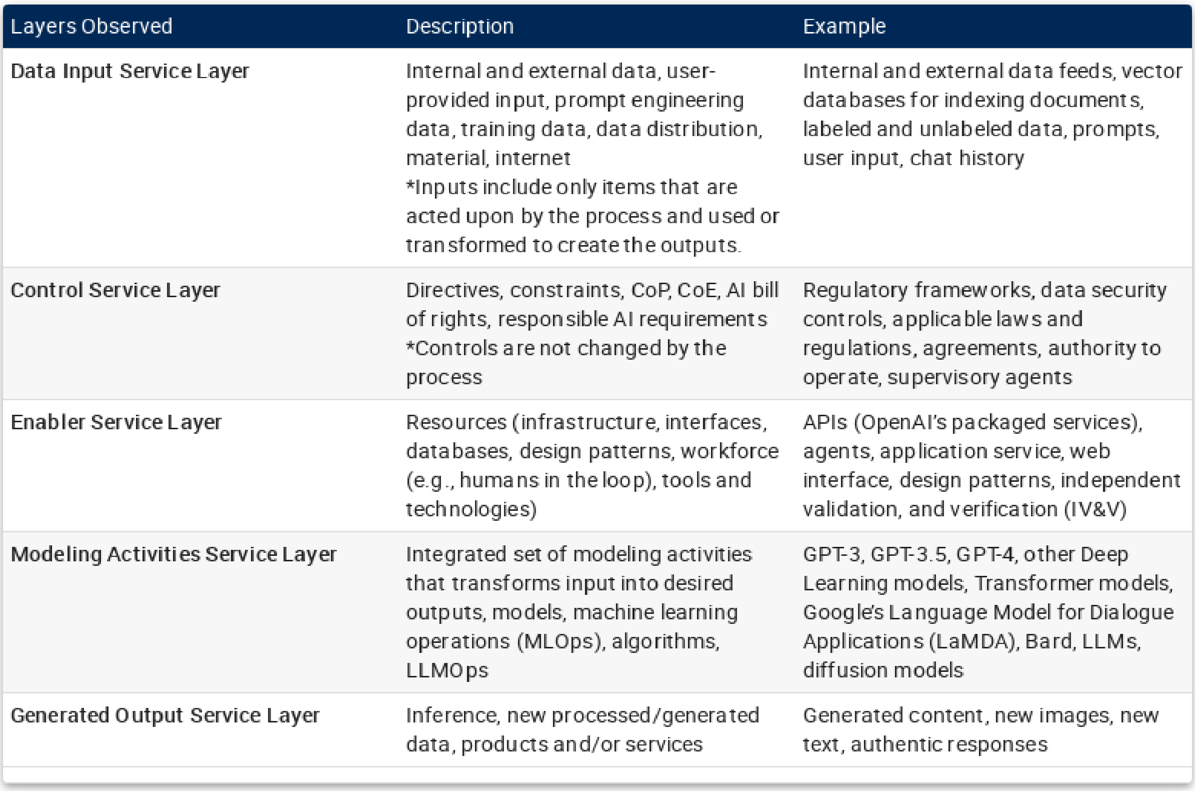


Source: Gartner
790301_C

Table 1 describes the layers in more detail.

Table 1: Highlighting AI Observability Layers for Generative AI Solutions

(Enlarged table in Appendix)



Layers Observed	Description	Example
Data Input Service Layer	Internal and external data, user-provided input, prompt engineering data, training data, data distribution, material, internet *Inputs include only items that are acted upon by the process and used or transformed to create the outputs.	Internal and external data feeds, vector databases for indexing documents, labeled and unlabeled data, prompts, user input, chat history
Control Service Layer	Directives, constraints, CoP, CoE, AI bill of rights, responsible AI requirements *Controls are not changed by the process	Regulatory frameworks, data security controls, applicable laws and regulations, agreements, authority to operate, supervisory agents
Enabler Service Layer	Resources (infrastructure, interfaces, databases, design patterns, workforce (e.g., humans in the loop), tools and technologies)	APIs (OpenAI's packaged services), agents, application service, web interface, design patterns, independent validation, and verification (IV&V)
Modeling Activities Service Layer	Integrated set of modeling activities that transforms input into desired outputs, models, machine learning operations (MLOps), algorithms, LLMOps	GPT-3, GPT-3.5, GPT-4, other Deep Learning models, Transformer models, Google's Language Model for Dialogue Applications (LaMDA), Bard, LLMs, diffusion models
Generated Output Service Layer	Inference, new processed/generated data, products and/or services	Generated content, new images, new text, authentic responses

Source: Gartner (August 2023)

Each layer interfaces with the modeling activities layer involved in supporting generative AI.

A layer consists of components. Each component is also a part of a solution pattern.

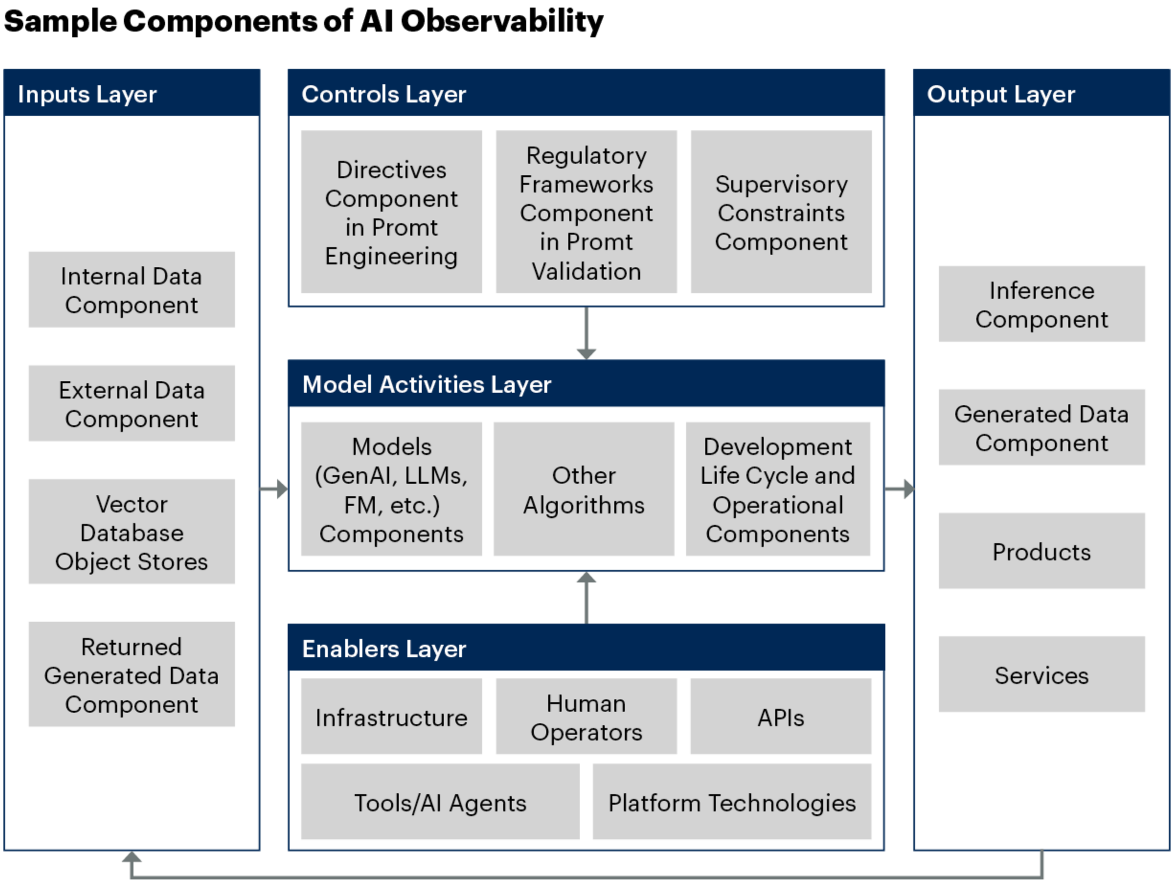
AI Observability Components

Components are meant to describe functionality or a set of functionalities that are intended to interact with other layers or components within the AI architecture.

> Observing components is a good way of measuring from "everywhere" and measuring the interactions throughout the solution architecture.

There is no firm set of components required to support AI observability. The purpose of highlighting observable components is to provide a framework for technical professionals to populate based on their solution requirements.Figure 8 provides sample components of AI observability and describes how each layer of components interacts with other layers of components within the AI solution.

Figure 8: Sample Components of AI Observability

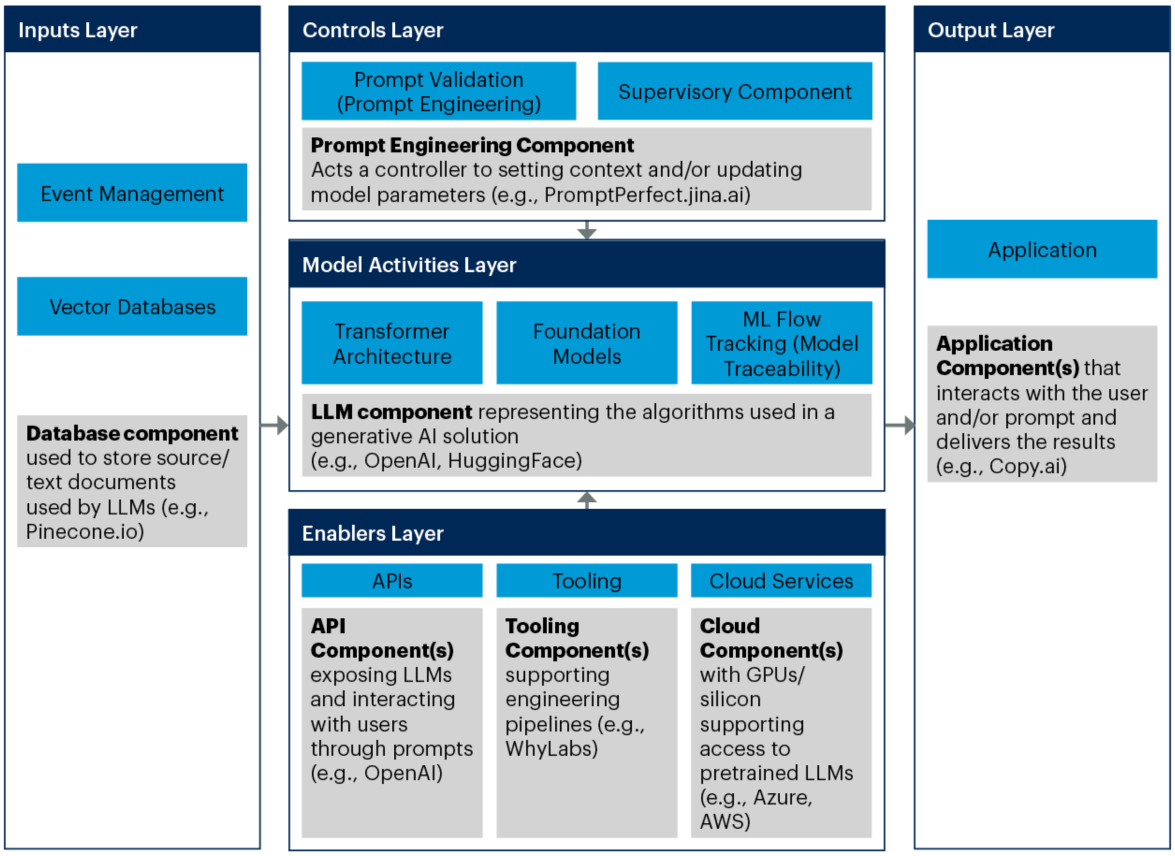

Sample Components of AI Observability
Inputs Layer
Internal Data Component
External Data Component
Vector Database Object Stores
Returned Generated Data Component
Controls Layer
Directives Component in Promt Engineering
Regulatory Frameworks Component in Promt Validation
Supervisory Constraints Component
Model Activities Layer
Models (GenAI, LLMs, FM, etc.) Components
Other Algorithms
Development Life Cycle and Operational Components
Enablers Layer
Infrastructure
Human Operators
APIs
Tools/AI Agents
Platform Technologies
Output Layer
Inference Component
Generated Data Component
Products
Services

Source: Gartner
790301_C

Note the flow of information from component to component and how they interact with modeling activities and produce outputs.For example, data components are composed of data sources that are used by AI modeling activities. Controls components influence the modeling activities. Enablers are the tools and technologies used to support the modeling activities. Modeling activities and machine learning processes are used to develop a generated output.To illustrate how the layers and components of AI observability can be used, Figure 9 provides a sample architectural pattern used to support an LLM application. There are numerous configurations of a solution. The point of this illustration is to demonstrate how an AI architect can design a generative AI solution architecture using the framework outlined in this research based on a requirement to influence or guide the generated AI output.

Figure 9: AI Observability for a Sample Recursive Generative Pretrained Transformer (GPT) System

AI Observability for a Sample Recursive Generative Pretrained (GPT) System



Inputs Layer

Event Management

Vector Databases

Database component used to store source/ text documents used by LLMs (e.g., Pinecone.io)

Controls Layer

Prompt Validation (Prompt Engineering)

Supervisory Component

Prompt Engineering Component Acts a controller to setting context and/or updating model parameters (e.g., PromptPerfect.jina.ai)

Model Activities Layer

Transformer Architecture

Foundation Models

ML Flow Tracking (Model Traceability)

LLM component representing the algorithms used in a generative AI solution (e.g., OpenAI, HuggingFace)

Enablers Layer

APIs

Tooling

Cloud Services

API Component(s) exposing LLMs and interacting with users through prompts (e.g., OpenAI)

Tooling Component(s) supporting engineering pipelines (e.g., WhyLabs)

Cloud Component(s) with GPUs/ silicon supporting access to pretrained LLMs (e.g., Azure, AWS)

Output Layer

Application

Application Component(s) that interacts with the user and/or prompt and delivers the results (e.g., Copy.ai)

Source: Gartner
790301_C

In the example in Figure 9, a use case for a recursive generative pretrained (GPT) system requires the fine-tuning of a foundational model to support an organization's context. For example, internal documents used to fine-tune a LLM or to enable prompts to be augmented with internal/proprietary data (also known as retrieval augmented generation [RAG]) are represented in a vector database component. Prompt validation is represented as a controller to guide the model as a part of fine-tuning the model used. Technology tooling and cloud services are represented as enablers, and application components are represented as the output.

AI Observability Interfaces

The components of AI observability use interfaces to communicate the flow of information from layer to layer. AI observability interfaces act as proxies that govern the information flow between components.

Interfaces play a critical role in helping technical professionals and business stakeholders understand and communicate the interaction between the components. They are used to manage the flow of information and allow for human-in-the-loop influence from layer to layer.There are three key benefits of using the interfaces within your AI observability reference architecture:

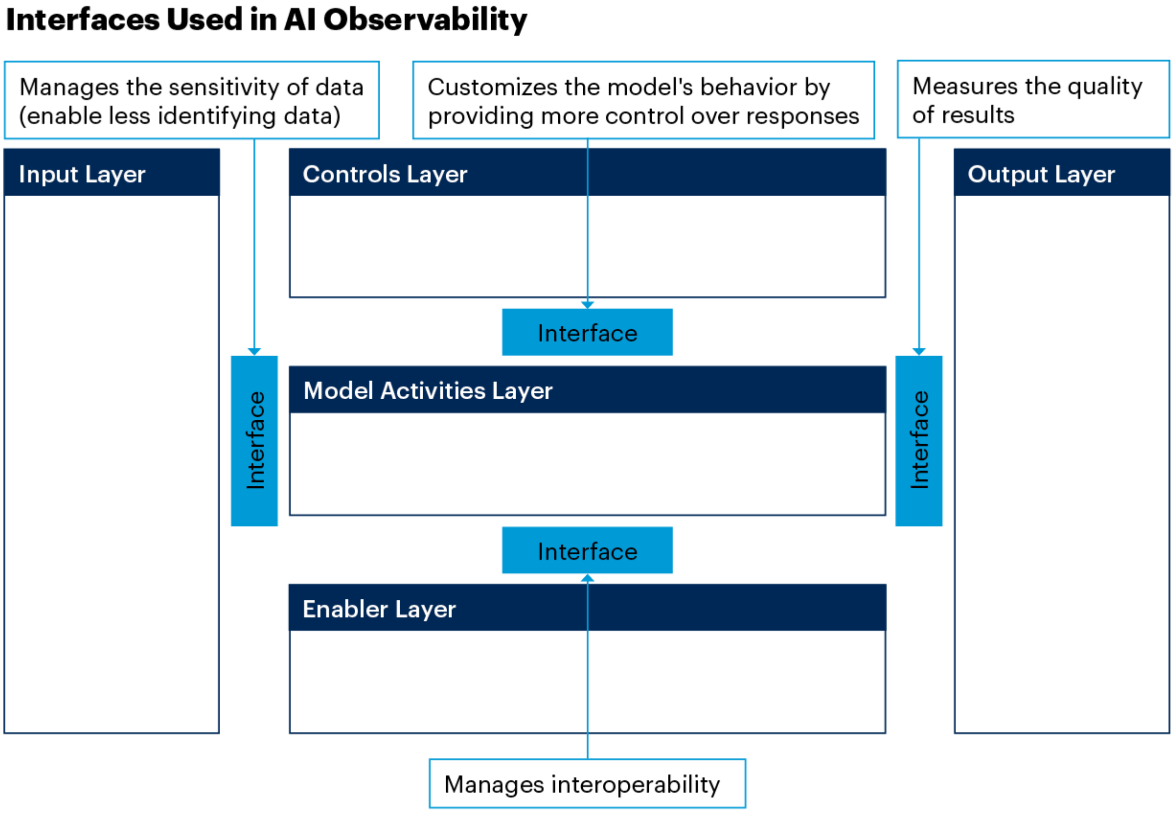
- **The ability to define the behavior that can be implemented by the AI models used in your AI solutions.** For example, data components in the Input layer can influence models. Regulatory framework components identified in the Controls layer can influence model behavior by putting constraints on a transformer's process of sequential input data.

- **By implementing interfaces, business and technical professionals can define methods for validating and verifying how information is consumed or produced by the modeling activities.** For example, a technology component like a cloud resource used to host a set of models in the operations of the AI solutions can be tested as a part of the system without interrupting other components in the system.

- **Interfaces allow for the collection of performance data and metrics to drive insights on model behavior and allow for a more integrated approach with modeling performance management platforms.** For example, technical professionals can use out-of-the-box model management platforms to capture telemetry data, logs, metadata from interfaces to better understand model behavior.

Figure 10 describes how the interfaces are used in the AI observability reference architecture.

Figure 10: Interfaces Used in AI Observability


Interfaces Used in AI Observability
Manages the sensitivity of data (enable less identifying data)
Customizes the model's behavior by providing more control over responses
Measures the quality of results
Input Layer
Controls Layer
Output Layer
Interface
Interface
Model Activities Layer
Interface
Interface
Enabler Layer
Manages interoperability

Source: Gartner
790301_C

The concept of interfaces is not new. Software engineers have been using interfaces for decades. However, interfaces that are used as methods to manage the interaction with generative AI models in support of AI observability are fairly novel. Some Gartner clients are finding success leveraging interfaces to control the information flow between components and layers.

Another reason to use interfaces is to build an observability layer to capture insights from all components in your AI solutions. There are numerous low-code technology platforms that support the visualization and analysis of observability data from interfaces. In fact, the following technology platforms are offering focused telemetry services capable of consuming data from interfaces to gain insights.

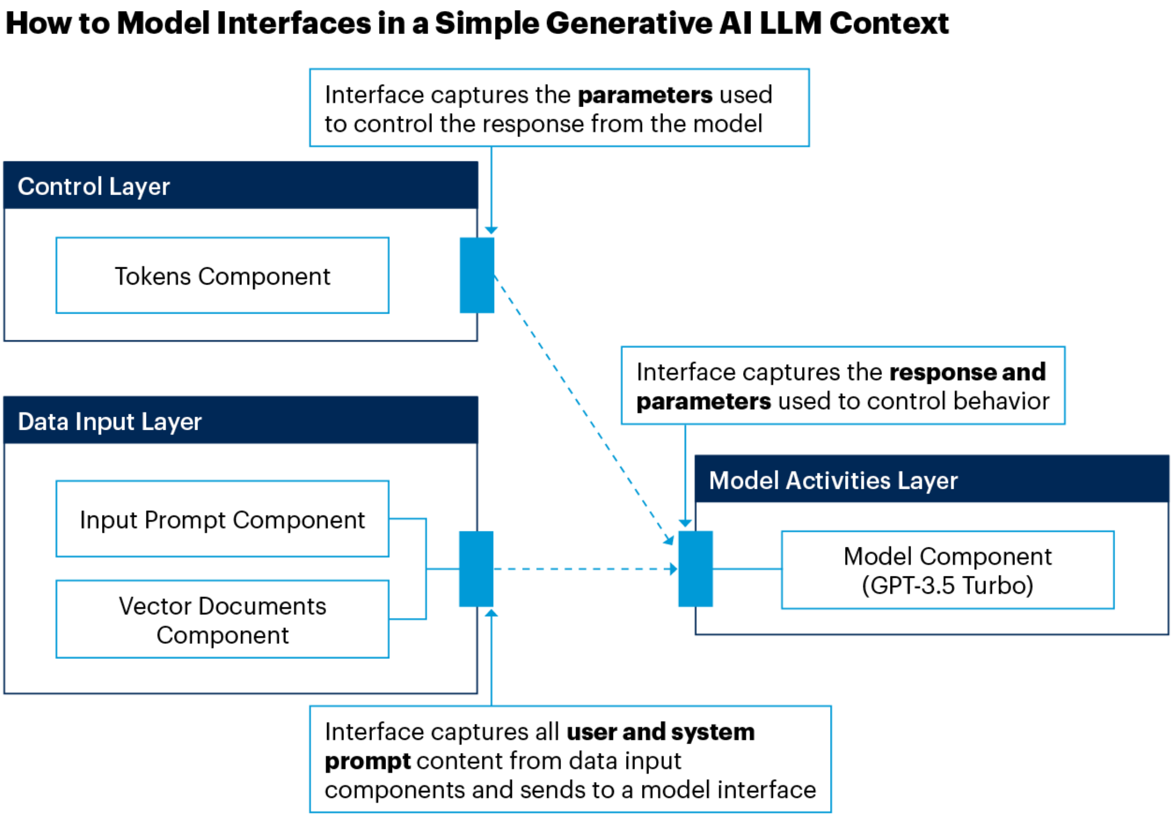
- **WhyLabs** focuses on ingesting data from interfaces from a ML modeling pipeline.
- **Fiddler** allows interfaces to populate their model performance management platform.

- **Splunk** is being customized to support ingesting models and performance data using interfaces.

Interfaces also encapsulate information flow data like methods, parameters and exchanged data flows needed to call other components in other layers. They can be aligned to common APIs, or they can be customized logic designed to manage the exchange of information between components and layers.

Figure 11 illustrates how to model interfaces in a simple generative AI LLM context.

Figure 11: Interface Example for LLM

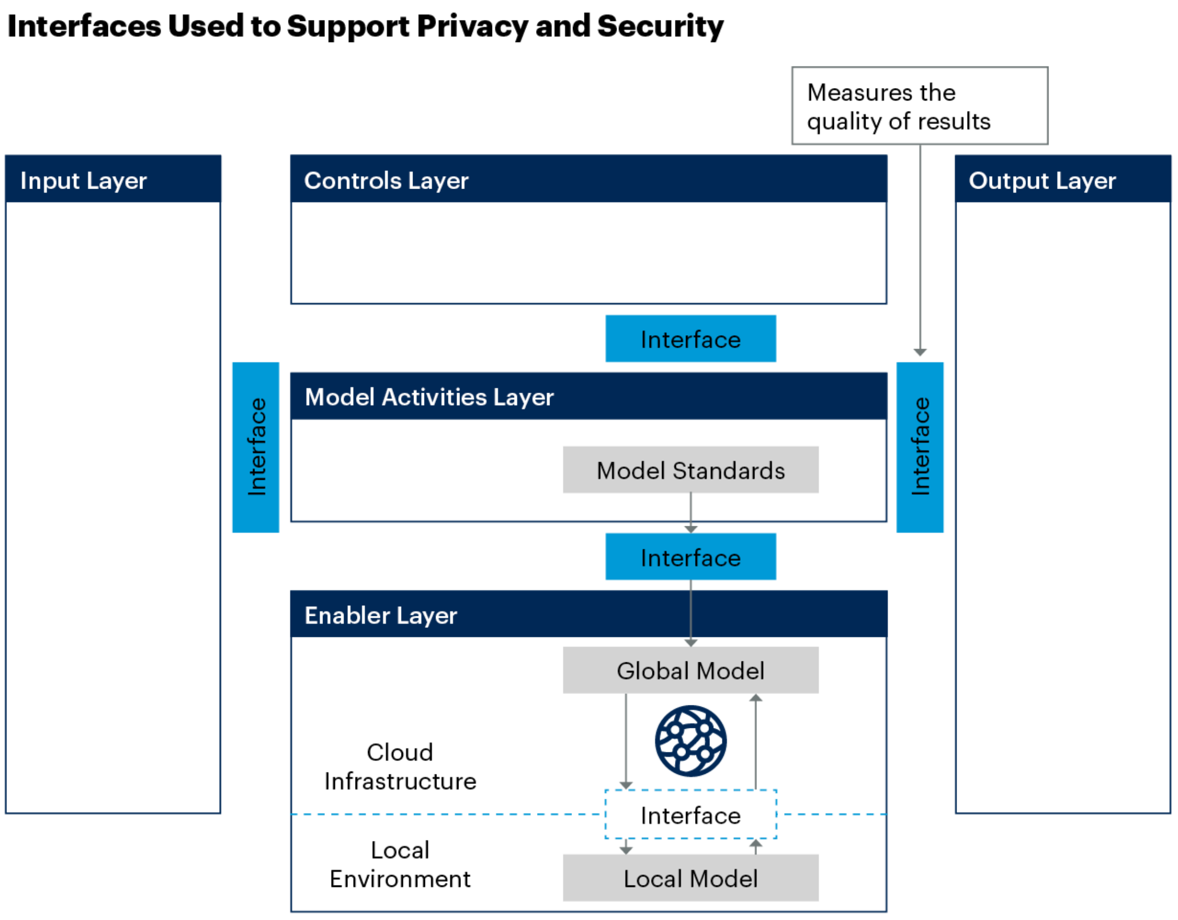

How to Model Interfaces in a Simple Generative AI LLM Context
Interface captures the **parameters** used to control the response from the model
Control Layer
Tokens Component
Interface captures the **response and parameters** used to control behavior
Data Input Layer
Input Prompt Component
Vector Documents Component
Model Activities Layer
Model Component (GPT-3.5 Turbo)
Interface captures all **user and system prompt** content from data input components and sends to a model interface

Source: Gartner
790301_C

Addressing Data and Model Privacy With Interfaces

AI observability interfaces are also being used to address data and model privacy in generative AI initiatives. For example, interfaces can manage requests and determine privacy strategies between federated models that span local infrastructure and global models from cloud services. Figure 12 highlights how an interface is used as a coordinator between models that span local and cloud infrastructure.

Figure 12: Interfaces Used to Support Privacy and Security



Source: Gartner
790301_C

Interfaces can manage model parameters and data that is shared between a global model and highly distributed local models. This technique is commonly found in federated ML use cases.

AI Observability Repository

An observability repository captures insights from the components and interfaces of the AI observability framework. Observability in the most traditional sense remains central to collecting traces, metrics and logs in the IT architecture. An AI observability repository contextualizes those insights to evaluate GenAI system health.

Many tools and vendor technology are available to support the development of a repository for insights. For example, the Databricks Lakehouse Platform and the recent acquisition of MosaicML offers a fruitful solution for an observability repository. In this repository, architects can deliver GenAI solutions that offer "a fast way to retain control, security, and ownership over data and insights" (see Databricks Signs Definitive Agreement to Acquire MosaicML, a Leading Generative AI Platform, Databricks).

> **The goal of an observability repository is to collect data from each component of the observability framework so that insights can be available to stakeholders.**

AI architects can add a semantic layer to the architecture to help contextualize the insights to various stakeholder perspectives. The combination of an observability layer with a semantic layer lets end users:

1. **Curate an inventory of data and model observability metrics.** This capability collects telemetry data from multiple domains at scale. There are several vendor platforms that support this capability. Gartner recommends leveraging data observability software to accelerate this capability. (For more information on data observability, see Deliver Trust by Adopting Data Observability Practices).

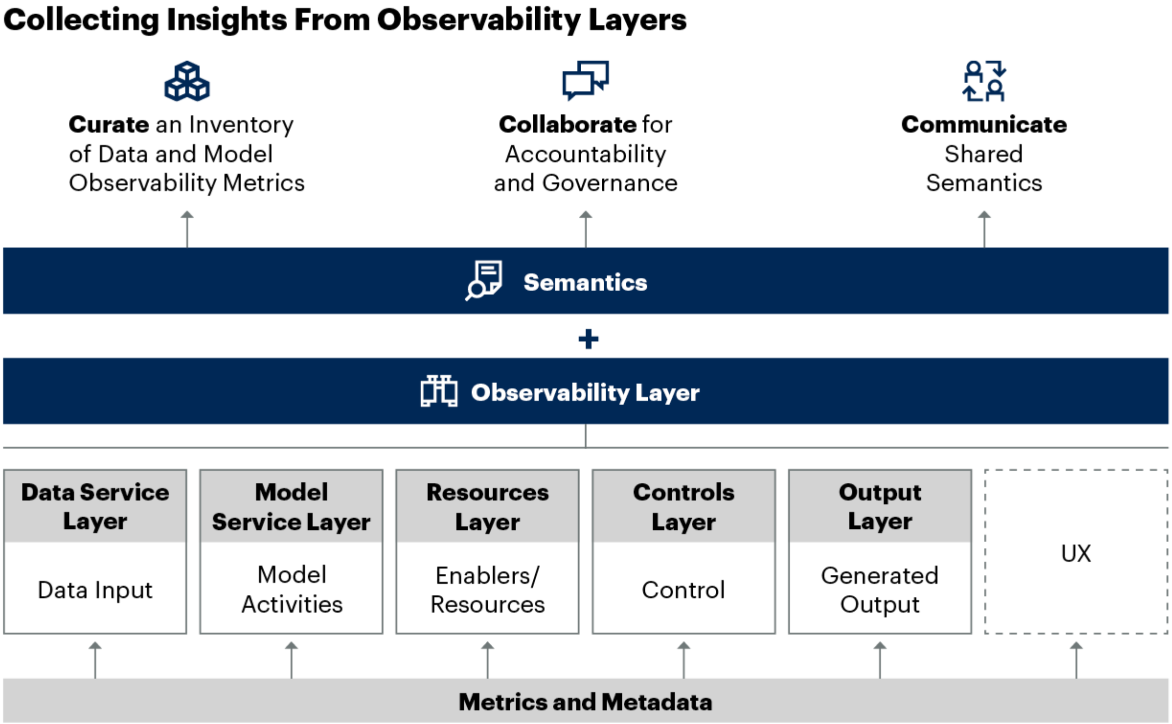
For model observability metrics, software companies like Fiddler provide model-centric observability instruments to enable monitoring and explainability of the models fueling generative AI solutions.

2. **Collaborate for accountability and governance.** This capability enables stakeholders to use insights from observability metrics to determine the expected results from data and models within generative AI life cycles. These metrics can be checked manually or automated and integrated into business workflows so all relevant stakeholders are informed of insights impacting the business. For example, many organizations establish one or more centers of excellence to govern the internal rules that support prompt functions for ChatGPT solutions. Some have found success building a registry of prompt functions that are encapsulated in the models, with details about how the prompt interacts in a business workflow.

3. **Communicate shared semantics.** This capability extends collaboration for accountability and governance to create a shared understanding of the insights from data and models that support generative AI across the organization or agency. For example, organizations use shared semantics to publish the resources used in training datasets to avoid using copyrighted data.

Figure 13 highlights how observability layers may be used to support the management of insights from the AI observability components.

Figure 13: Collecting Insights From Observability Layers



Source: Gartner
790301_C

Capturing the User Experience in Observability Repositories

A relatively new feature offered by technology providers is the acquisition of user experience (UX) data. Capturing UX data enables greater design for generative AI experiences. The Observability layer outlined in Figure 13 can be extended to include data from the user experience interacting with your generative AI solution.

For example, human-to-computer interaction experts are tapping into observability repositories to discover how to develop more personalized experiences as enablers. They also determine how flagging AI-generated content alters UX.

Differentiating AI Observability

How Is AI Observability Different From Other Frameworks?

> **What makes AI observability different is that it's not about controlling generative AI output, but about provoking it. What other architectural considerations enable nudging these systems to avoid unintended consequences?**

Numerous frameworks to support the oversight of generative AI are maturing. However, many of the existing frameworks are incomplete, descriptive and not proactive enough or are too theoretical for technical professionals to use in their architectural patterns when designing GenAI solutions.

For example, explainable AI provides tools and frameworks to help interpret AI output (see Innovation Insight for Bias Detection/Mitigation, Explainable AI and Interpretable AI). It focuses on understanding input data, modeling activities and generated output. However, it does not implement controls and technology constraints that help influence generative AI models to prevent unintended consequences. There are explainable AI methods that AI architects use to explain the model and output in a solution architecture business layer, and newer premodeling explainability methods have recently been introduced. However, these methods still lack controllability.Similarly, ModelOps, MLOps and model monitoring also focus on input data, modeling activities, and the generated output (see Use Gartner's MLOps Framework to Operationalize Machine Learning Projects and Launch an Effective Machine Learning Monitoring System). These model-centric frameworks deftly manage the full life cycle of ML models. They can also be ingrained in the solution architecture as descriptive analytics to better understand how the models are performing. However, these frameworks remain limited in their ability to establish proactive controls either through the data or through the technology enablers to influence the behavior of the models or application hosting generative AI solutions.

As a part of model management, Gartner recognizes AI trust, risk and security management (AI TRiSM) (see Top Strategic Technology Trends for 2023: AI Trust, Risk and Security Management). AI TRiSM supports AI model governance, trustworthiness, fairness, reliability, robustness, efficacy and data protection. It includes solutions, techniques and processes for model interpretability and explainability, AI data protection, model operations and adversarial attack resistance. However, AI TRiSM falls short of controllability and influence over enablers that support the model. For example, AI TRiSM would not support the supervision of data used to influence commercially available API services that use pretrained deep learning models for GenAI solutions.

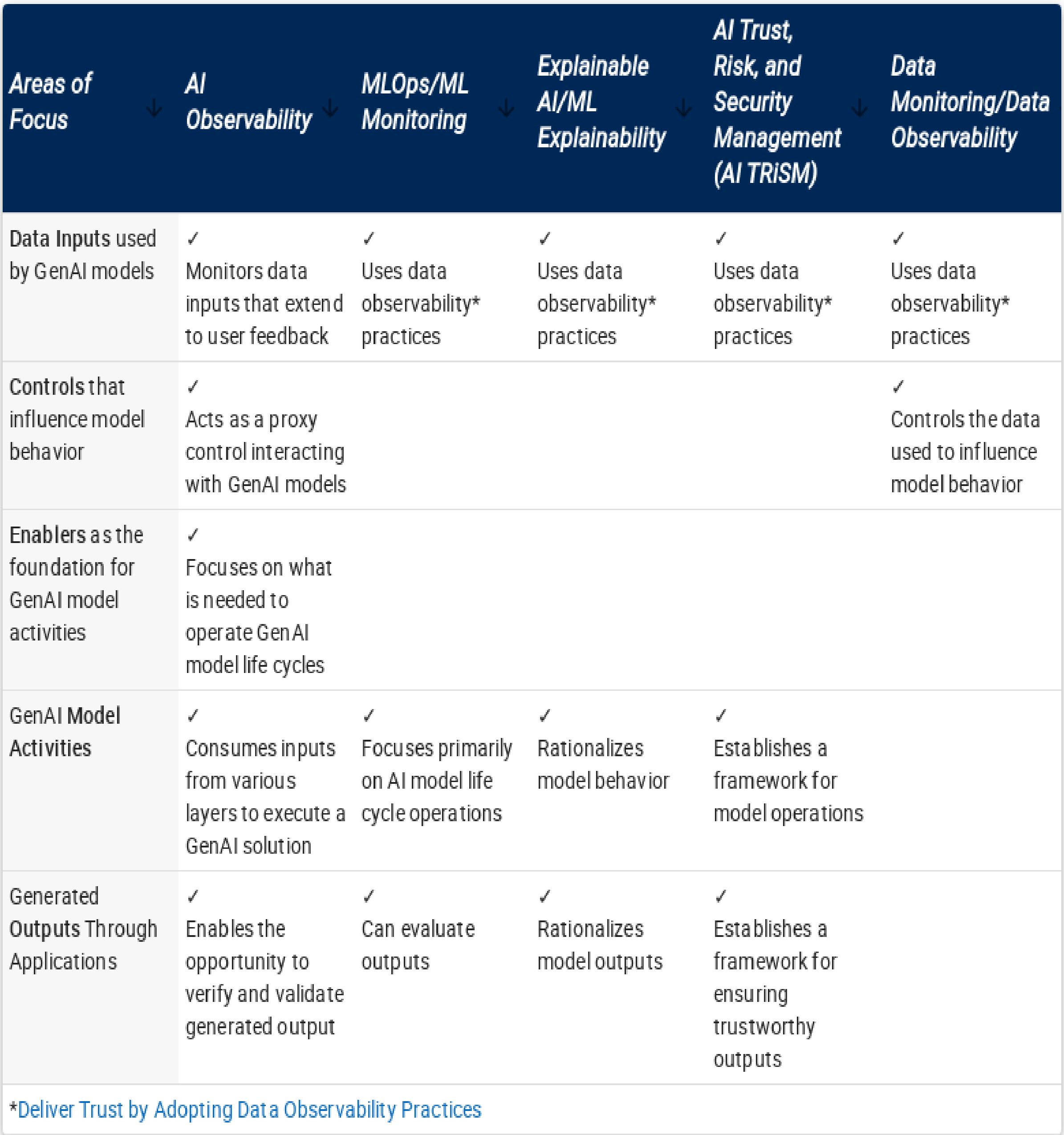
It is also important to recognize the similarities in the practice of responsible AI (see A Comprehensive Guide to Responsible AI).

> Organizations often struggle to differentiate the practices of responsible AI and AI observability. The distinction is in the implementation. AI observability focuses more on instrumenting your generative AI solution so that you gain influence over the model behavior. Responsible AI focuses on designing and deploying AI with good intentions.

However, responsible AI requires contextualization because it may mean something different to different stakeholders or end users. Implementing AI observability should be seen as a method for establishing the start of responsible AI. Table 2 provides a brief overview of the differences in some of the most common frameworks based on the AI observability layers described in this research. The checkmark indicates that the framework supports the area of focus.

Table 2: Differentiating AI Observability from Other Frameworks

(Enlarged table in Appendix)



Areas of Focus	AI Observability	MLOps/ML Monitoring	Explainable AI/ML Explainability	AI Trust, Risk, and Security Management (AI TRiSM)	Data Monitoring/Data Observability
Data Inputs used by GenAI models	✓ Monitors data inputs that extend to user feedback	✓ Uses data observability* practices	✓ Uses data observability* practices	✓ Uses data observability* practices	✓ Uses data observability* practices
Controls that influence model behavior	✓ Acts as a proxy control interacting with GenAI models				✓ Controls the data used to influence model behavior
Enablers as the foundation for GenAI model activities	✓ Focuses on what is needed to operate Gen AI model life cycles				
GenAI Model Activities	✓ Consumes inputs from various layers to execute a GenAI solution	✓ Focuses primarily on AI model life cycle operations	✓ Rationalizes model behavior	✓ Establishes a framework for model operations	
Generated Outputs Through Applications	✓ Enables the opportunity to verify and validate generated output	✓ Can evaluate outputs	✓ Rationalizes model outputs	✓ Establishes a framework for ensuring trustworthy outputs	

*Deliver Trust by Adopting Data Observability Practices

Source: Gartner (August 2023)

Data observability offers a good starting point in designing for key insights from the Data Input layer because it handles many aspects of that layer. However, it has a limited ability to monitor high-dimensional or complex data types like images, audio and video. Although it doesn't focus on model insights, it is highly complementary to many of the frameworks described above and is complementary to AI observability. (For more information on data observability, please refer to Deliver Trust by Adopting Data Observability Practices).

Architecting for AI Observability

How Do Technical Professionals Architect AI Observability for Generative AI Solutions?

Effective AI observability for generative AI is based on two critical AI model-centric capabilities:

- Influencing the AI model activities via a modular approach, and
- Monitoring AI model behavior through some form of instrumentation.

Both capabilities are embedded in well-architected technical frameworks.

Gartner clients architecting AI observability for generative AI leverage the modular open system approach (MOSA) and apply the approach to the framework outlined in this research. MOSA works well for architecting AI observability, especially when generative AI is used to build software, like in code assistant tools (e.g., GitHub Copilot, Amazon CodeWhisperer). MOSA is primarily used in software engineering.

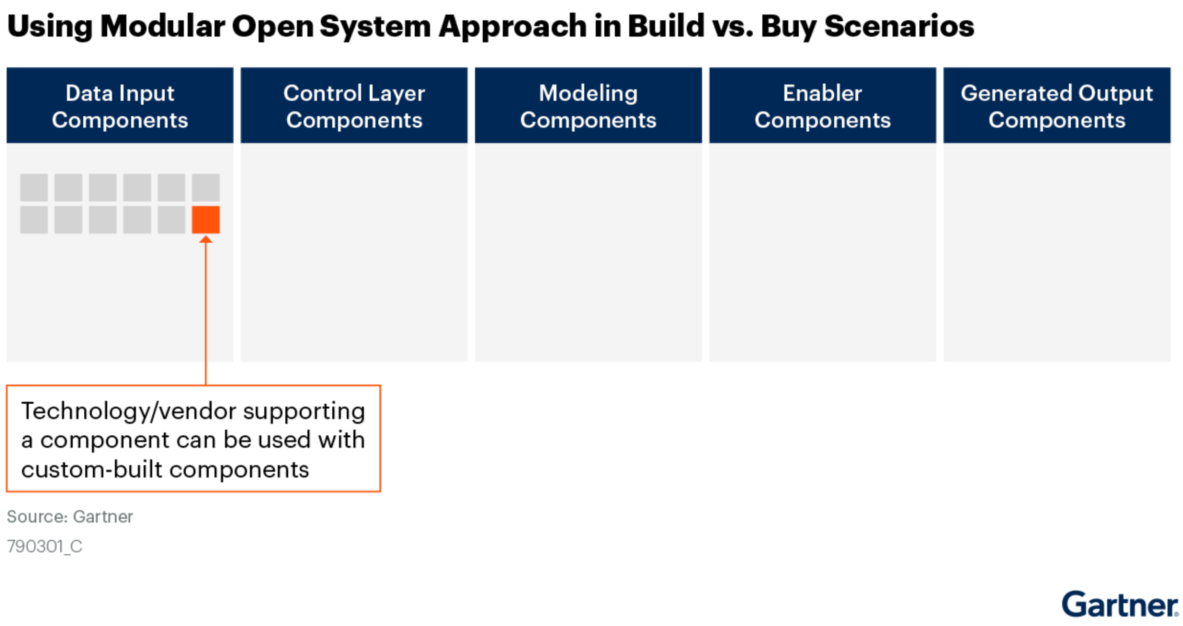
Modular Open System Approach (MOSA)

MOSA is an architectural approach that allows for components to be replaced or added throughout the life cycle of a system. It is used primarily within the U.S. Department of Defense (DOD) to support system acquisitions. However, private sector organizations have found success leveraging MOSA to instrument observability within their generative AI architectures.

Part of the success of using MOSA in AI observability is in the ability to architect both build vs buy implementation strategies. In other words, part of an AI observability framework can be purchased off the shelf while other parts are customized to support a specific capability that is unique to an organization's requirements. For example, an organization can purchase a model performance management component, but build its own control components. It can use the purchased model performance management system to determine if the control components are properly influencing the model.

Figure 14 illustrates how MOSA is used to implement components in a build-versus-buy scenario.

Figure 14: Using Modular Open System Approach in Build vs. Buy Scenarios


Using Modular Open System Approach in Build vs. Buy Scenarios
Data Input Components
Control Layer Components
Modeling Components
Enabler Components
Generated Output Components
Technology/vendor supporting a component can be used with custom-built components
Gartner

Source: Gartner
790301_C

Using MOSA to design AI observability into your solution architecture offers extended advantages:

- **Describing observability components and interfaces in a solution architecture captures explicit knowledge about how information is flowing through the AI system,** even when there is limited knowledge of the models' internal workings.
- **Introducing a modular design as a method of instrumentation helps establish well-defined, cohesive and traceable functionality and functional decomposition.** This is particularly important when dealing with high-fidelity generative AI models that are used to contribute legacy code modernization or design strategies for complex systems (e.g., transportation systems).
- **The ability to design to open standards.** Pervasive frameworks offering responsible practices for AI and collective action are aggressively ushering in open standards for AI (see Partnership on AI's (PAI's) Responsible Practices for Synthetic Media). This provides an opportunity for technical professionals to translate and incorporate those standards into the architectural design strategy using MOSA and the framework outlined in this research.

The majority standard for implementing MOSA for AI observability during design is using unified modeling language (UML) or SysML. For more information on using UML, see Data Modeling to Support End-to-End Data Architectures.

As previously mentioned, one of the most common use cases for architecting AI observability is in software engineering because generative AI is commonly used to build software. Using generative AI to build software is a growing trend. In a recent Gartner survey, more than half of respondents expect generative AI will be used to build software (see Quick Answer: How Can Generative AI Tools Speed Up Software Delivery?).

MOSA works well for building software and architecting for observability because it enables AI architects and AI engineers the ability to describe the layers, components/modules, and interfaces needed to support AI observability.

Recommendations

Technical professionals tasked with designing architectures to support generative AI solutions should focus on applying observability as a capability within the architecture. Several frameworks overlap with AI observability principles, and many of these frameworks fall under technical architecture in the broader context of AI observability. Well-engineered and implemented observability solutions will provide valuable insights needed for evaluating the behavior of generative AI and providing a capability to influence the generated output.

- Build observability into your generative AI solution by including components, interfaces and repositories that reflect the AI observability principles into your AI solution architectural pattern. AI observability is a solution architectural pattern for generative AI systems to establish a practice for designing and implementing generative AI to meet the functional and nonfunctional requirements of the system.
- Expand other observability frameworks to include the components needed to support generative AI solutions. For example, if you are leveraging data observability, then expand the data observability framework to include model controls and enablers.
- Leverage enterprise model performance management platforms to enable stakeholders to monitor, explain, analyze and improve generative AI model performance. There are several out-of-the-box technology platforms available to help accelerate the ability to capture insights from generative AI model behavior. Use these platforms as insight engines and build controls around them. For example, technology vendors like Fiddler offer interactive explainable AI capabilities that are customizable and complementary to broader AI observability frameworks.

Establish oversight services to component interfaces of AI observability by assigning independent verification and validation resources to all interfaces and components of the AI observability architecture framework. For example, each component interface provides an opportunity for independent verification and validation.

Conclusion

AI observability is a start to managing and governing generative AI solutions. Despite many overlapping frameworks attempting to address the risks of generative AI technologies, AI observability provides a more holistic approach to ensuring generative AI is manageable from an architectural perspective. Technical professionals looking for a more practical approach to implementing observability and controllability should consider the AI observability framework outlined in this research.

However, AI observability is just a starting point. There are several areas of improvement that AI observability doesn't address in supporting generative AI solutions. Technical professionals implementing AI observability as a solution architecture for generative AI should consider improving:

- How to describe and detect when regulatory frameworks that are modeled as controls are violated by generative AI models
- How to implement behavioral models that predict how generative AI models will behave
- How to instrument AI agents interacting with other AI agents for observability

The market for AI observability technology solution providers is rapidly emerging as a key ingredient for avoiding some of the key risks associated with the deployment of generative AI. It continues to offer a viable architectural solution that technical professionals can implement into their existing AI architectures.

However, many of the top-tier cloud service providers, like Amazon Web Services (AWS), Microsoft Azure and Google are behind in their ability to instrument AI observability within their own public services.

Gartner is encouraging service providers to instrument greater AI observability into their solutions. In the meantime, AI solution architects will need to design for the principles outlined in this research and implement into their design patterns.

Gartner clients continuously ask about the integration of observability features in current major players involved in providing generative AI solutions (e.g., Google's Pathways Language Model [PaLM)], Amazon/Hugging Face LLMs or IBM's foundation models). At the time of publishing this research, there has not been an indication of how these players will instrument their generative AI solutions for more in-depth AI observability. However, we are optimistic that greater support for AI observability is not far away.

Acronym Key and Glossary Terms

Acronym	Definition
ChatGPT	A generative AI technology that is open and available for public use.
OpenAI APIs	Commercially available APIs from the company OpenAI
Large language model (LLM)	A large language model (LLM) is a specialized type of artificial intelligence (AI) that has been trained on vast amounts of text to understand existing content and generate original content.
Azure OpenAI Service	Azure OpenAI Service provides API access to OpenAI's powerful language models, including the GPT-3, Codex and embeddings model series. In addition, the new GPT-4 and ChatGPT (GPT-35-Turbo) model series have now reached general availability.
GPT	Generative pretrained transformers are a series of neural network models that use the transformer architecture to generate output.
GPT-4	GPT-4 is a powerful large language model (LLM) with generative AI capabilities developed by OpenAI. GPT-4 uses a transformer model architecture to predict the next token (a part of a word). It was trained on large amounts of public and some private data, and the model was fine-tuned using reinforcement learning with human feedback (RLHF). The key new or improved capabilities in GPT-4 include the ability to support image and text inputs, as well as support for longer forms of input and output.
MOSA	Modular open system approach; a technical and business strategy for designing an affordable and adaptable system.
Transformer architecture	A type of deep neural network architecture that computes a numerical representation of training data.

Evidence

2023 Gartner Voice of the Client Content Survey. This survey was conducted online with 820 engaged Gartner clients in IT and business leader roles from 9 May to 31 May 2023. The objective of the survey was to better understand client needs, and to gauge use and expectations of generative AI in their organizations. Participants represented a wide range of industries and came from across the world: 56% from North America, 27% from EMEA, 13% from APAC and 4% from Latin America. All participants had recently engaged with Gartner's content on gartner.com (within the last 90 days).

Recommended by the Author

Some documents may not be available as part of your current Gartner subscription.

Deliver Trust by Adopting Data Observability Practices

Hype Cycle for Monitoring and Observability, 2023

Table 1: Highlighting AI Observability Layers for Generative AI Solutions

Layers Observed	Description	Example
Data Input Service Layer	Internal and external data, user-provided input, prompt engineering data, training data, data distribution, material, internet *Inputs include only items that are acted upon by the process and used or transformed to create the outputs.	Internal and external data feeds, vector databases for indexing documents, labeled and unlabeled data, prompts, user input, chat history
Control Service Layer	Directives, constraints, CoP, CoE, AI bill of rights, responsible AI requirements *Controls are not changed by the process	Regulatory frameworks, data security controls, applicable laws and regulations, agreements, authority to operate, supervisory agents
Enabler Service Layer	Resources (infrastructure, interfaces, databases, design patterns, workforce (e.g., humans in the loop), tools and technologies)	APIs (OpenAI's packaged services), agents, application service, web interface, design patterns, independent validation, and verification (IV&V)
Modeling Activities Service Layer	Integrated set of modeling activities that transforms input into desired outputs, models, machine learning operations (MLOps), algorithms, LLMOps	GPT-3, GPT-3.5, GPT-4, other Deep Learning models, Transformer models, Google's Language Model for Dialogue Applications (LaMDA), Bard, LLMs, diffusion models
Generated Output Service Layer	Inference, new processed/generated data, products and/or services	Generated content, new images, new text, authentic responses

Source: Gartner (August 2023)

Table 2: Differentiating AI Observability from Other Frameworks

Areas of Focus ↓	*AI Observability* ↓	*MLOps/ML Monitoring* ↓	*Explainable AI/ML Explainability* ↓	*AI Trust, Risk, and Security Management (AI TRiSM)* ↓	*Data Monitoring/Data Observability* ↓
Data Inputs used by GenAI models	✓ Monitors data inputs that extend to user feedback	✓ Uses data observability* practices	✓ Uses data observability* practices	✓ Uses data observability* practices	✓ Uses data observability* practices
Controls that influence model behavior	✓ Acts as a proxy control interacting with GenAI models				✓ Controls the data used to influence model behavior
Enablers as the foundation for GenAI model activities	✓ Focuses on what is needed to operate GenAI model life cycles				
GenAI **Model Activities**	✓ Consumes inputs from various layers to execute a GenAI solution	✓ Focuses primarily on AI model life cycle operations	✓ Rationalizes model behavior	✓ Establishes a framework for model operations	

Areas of Focus	*AI Observability*	*MLOps/ML Monitoring*	*Explainable AI/ML Explainability*	*AI Trust, Risk, and Security Management (AI TRiSM)*	*Data Monitoring/Data Observability*
Generated **Outputs** Through Applications	✓ Enables the opportunity to verify and validate generated output	✓ Can evaluate outputs	✓ Rationalizes model outputs	✓ Establishes a framework for ensuring trustworthy outputs	

*Deliver Trust by Adopting Data Observability Practices

Source: Gartner (August 2023)