BEST PRACTICE REPORT

# The Architect's Guide To Generative AI

Prepare For Multiple Models With Multiple Modes, Doing Many Tasks In Many Use Cases

January 12, 2024

By Brian Hopkins, Charles Betz, Jeff Pollard, Alvin Nguyen, Will McKeon-White with Srividya Sridharan, Rowan Curran, Carlos Casanova, J. P. Gownder, Mike Gualtieri, Martha Bennett, Ted Schadler, Diego Lo Giudice, James McGlynn, Demi Starks

FORRESTER®

## Summary

Generative AI (genAI) has become a boardroom topic, but the market has focused almost exclusively on large language models. The future of genAI is much bigger than question-and-answer-style chat. Architects must get involved, helping their organization understand its potential and the building blocks needed for a dizzying array of genAI-powered applications. Using patterns (like retrieval-augmented generation), views, and principles, architects can show stakeholders how all the technical elements work together. They can also help their business improve tradeoff decisions between risk and reward.

# Architects Have A Lot To Figure Out Beyond LLMs

ChatGPT turned AI into a boardroom conversation and the market quickly became obsessed with large language models (LLMs). In fact, the term LLM has become synonymous with generative AI (genAI). But it'll take much more than a single gigantic LLM and one public chat application to get genAI right. Most of the code in genAI-powered applications we studied had little to do with the LLM chosen. Some genAI applications won't need LLMs or diffusion models at all. Architects must understand this, or their firm will fail to deliver and differentiate. Today, technology architects grapple with:

- **A wide range of genAI deployment options and misplaced expectations.** Enterprises are deploying genAI in multiple ways, from BYOAI ("bring your own AI") to training custom models (see Figure 1). We see a wide range of expectations as well. Overconfident executives have experienced ChatGPT's magic and want technology teams to just move fast and get started. They don't understand that what looks simple for OpenAI is actually very difficult for specific knowledge domains, other use cases, and data protection requirements. Governance, security, and risk teams are far less thrilled with known issues of factual inaccuracy, irrelevant outputs, privacy, data leakage, intellectual property, and more, giving rise to some misplaced expectations of how to make the most of this opportunity.
- **Balancing a steep learning curve with the urge to move fast.** Risk management reluctance isn't stopping many firms; nor should it. The opportunity is simply too great, and the worse course of action is to wait and see. However, this leaves architects stuck setting realistic expectations and navigating risk management concerns. One data scientist bluntly told us, "Executives underestimate this hallucination problem, and there isn't really a good solution for it." Generative models are also evolving at a blinding pace. Just a few years ago, GPT-2 could barely string together coherent sentences. Now, many multimodal models are emerging — these aren't just language. GPT-4 can take images as inputs, and Google's Med-PaLM 2 recently answered US Medical Licensing Examination questions with 85% accuracy. Many of its learnings come from diagnostic data and imagery.
- **A big list of technology to-dos, some new and some old.** Many generative capabilities are common to all AI systems, but others aren't. You'll likely need new capabilities like vector databases, knowledge graphs, LLM model quantization

tools, and others. Many firms are also opting for private hosting environments to protect their data, rather than using multitenant genAI-as-a-service tools from OpenAI, Anthropic, or others. Lastly, the number of startups hawking new genAI and LLM tools has mushroomed. The term LLM Ops has exploded in messaging from established players like Databricks and seed round startups like Valohai. Architects will need to help their firms determine if any new tools are needed and, if so, what the gaps are.

### Figure 1

**Beyond BYOAI, There Are Three GenAI Acquisition Strategies**

| | Default<br>**BYOAI**<br>**Least advanced** | Option 1<br>**Embedded in other software packages** | Option 2<br>**Tune and engineer pretrained model** | Option 3<br>**Train custom model**<br>**Most advanced** |
|---|---|---|---|---|
| **Examples** | • Public ChatGPT<br>• Dall-E<br>• Bard<br>• Duet.ai | • Salesforce<br>• Microsoft 365<br>• Dynatrace<br>• Zebrium | • Forrester Izola (Llama 2)<br>• Wendy's FreshAI (Google BERT)<br>• Woods Art Insitute (OpenAI Dall-E) | Hippocratic AI |
| **How** | Do nothing, your employees are already doing it. | • Upgrade your software.<br>• Buy additional hardware if required.<br>• Feed the software your data.<br>• Tune model outputs as software allows. | • Choose a suite of pretrained models.<br>• Train and tune them.<br>• Enrich prompts, possibly via RAG.<br>• Create governance and learning loops.<br>• Deploy models and optimize costs. | • Choose a base model or build your own.<br>• Choose pretrained models for other tasks.<br>• Train and tune the models.<br>• Deploy model and optimize costs. |
| **Who's doing it** | Every business | Every business that buys packaged software | 71% of organizations have started experimenting with genAI. | Primarily high-tech firms |

Base: 275 AI decision-makers

Source: Forrester's September 2023 Artificial Intelligence Pulse Survey

# Tomorrow Will Bring Even More Models, Modes, Tasks, And Use Cases

GenAI's current state is enough to keep architects busy managing tradeoffs just to succeed. But you must prepare for a future that involves many more models, operating in multiple modes, across many application tasks. We studied dozens of genAI patterns and architectures and found they all have one thing in common — none have just one generative model. In fact:

- **GenAI models perform many tasks in applications.** Text-to-text language generation is just one task that a growing number of models perform, operating in multiple modes across many use cases (see Figure 2). You must prepare for an architecture that employs dozens of models selected from libraries of thousands.
- **Not all genAI models need to be large.** Smaller models tuned and engineered for narrow knowledge domains can be more desirable than gigantic, general purpose LLMs from Anthropic, Google, and OpenAI. You must prepare for an architecture that can support both very large public models for some tasks and smaller, tuned, and specialized models for others. For example, one data scientist we spoke with was investigating ChatGPT for one task in his architecture due to performance, even though his firm's LLM was a smaller, open-source model running on virtual private cloud.

**Figure 2**

You'll Need Many Models For Your GenAI-Powered Portfolio

| Model task | Mode | Examples | Typical use cases |
|---|---|---|---|
| Text-to-vector (embedding) | Text-only | BERT, OpenAI embeddings, BAAI general embedding | • Feature extraction<br>• Semantic similarity<br>• Answer extraction<br>• Document classification |
| Text-to-text | Text-only | GPT-3, InstructGPT, XL.net, T5 | • Summarization<br>• Synthetic data<br>• Content generation<br>• Coding assistance<br>• Validation/moderation |
| Text-to-code | Text-only | Codex, CodeBERT | • Code generation<br>• Code explanation<br>• Code documentation<br>• Unit test generation |
| Multilingual text-to-text | Text-only | mBERT, XLM | • Translation<br>• Localization |
| Image-to-text | Text and image | DePlot, CLIP, Midjourney | • Image classification<br>• Business intelligence<br>• Visual search<br>• Scientific research |
| Text/image-to-text | Text and image | GPT-4, PaLM 2, Llama 2 | • Most of the above |
| Document extraction | Text and image | LayoutLM | • Search<br>• Knowledge extraction |
| Text-to-image/AV | Text, image, and video | Dall-E, Stable Diffusion XL, CLIP | • Content generation |



# Create A Lingua Franca To Drive Stakeholder Understanding

Architects must balance short-term needs with long-term plans, but this can pull teams in different directions. For example, ChatGPT can answer questions by looking at web pages to update its knowledge, which may deliver what your business needs. However, architects looking at costs, privacy concerns, answer quality, and future needs may see advantages in platforms like Amazon Bedrock. How do you make the right decisions and explain these to your stakeholders? Focus on building blocks, patterns, and architecture that supports rapid experimentation.

## Start With A Vocabulary Of Architecture Building Blocks

The Open Group Architecture Framework (TOGAF) uses the term “architecture building blocks” to mean:

> “A basic package of functionality representing a specific aspect of an enterprise architecture. Building blocks serve as a fundamental unit of architecture design and development, describing the key elements of enterprise architecture and how they relate to one another.”

Define genAI building blocks as your lingua franca to promote a common understanding of needs and solutions. LLMs are the most common building block today, but you’ll need more to design even the simplest solution (see Figure 3).

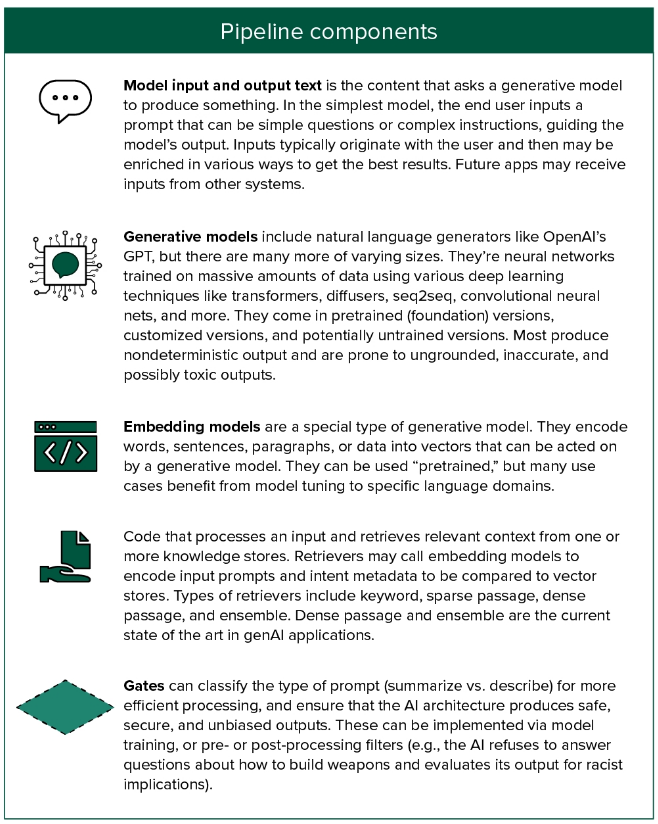
You’ll need different building blocks for different architecture views, such as solution, data, infrastructure, or security. The most common solution building blocks we found are organized into three types — general pipeline components, specialized governance gates, and intelligence and support services that are delivered from other layers in your architecture:

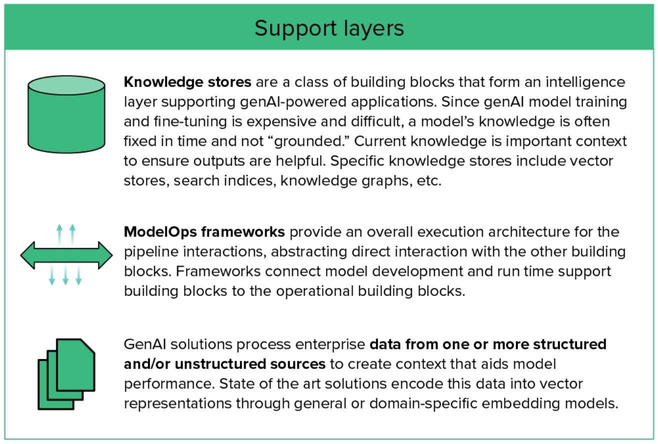
- **Pipeline building blocks include the models and orchestrations around them.** These building blocks describe the overall application workflow. They include branching logic to capture, source, and encode content for model processing, and flow the information through gates to be filtered, classified, and sent to the right generative models. Pipeline components also handle services calls to intelligence layers and engineer prompts for each generative model across various modes like text, data, and imagery.
- **Gate building blocks govern model operations and provide feedback.** Control gates in your pipeline play both an input shaping and governance role. We see two general types of gates: Intent gates examine and shape input content as it’s fed to a generative model to identify various question intents and route them appropriately. They can also look for hackers’ nefarious use of language that can circumvent model grounding, causing harmful outputs. Governance gates monitor and learn about what comes out. High-tech services firms like West Monroe are building feedback loops in governance gates that call a service that evaluates response accuracy and toxicity. Firms are also experimenting with using specially trained generative models for governance checks as part of output and intent gates.
- **Layers of intelligence and support provide reliable access to critical services.** Services embedded in layers of intelligence surface content and data to your

genAI-powered applications. Language-generation models read in prompt-enriched text as context for answering questions. Common technologies exposed in these layers include vector data stores and enterprise knowledge graphs, which applications use to enrich prompts with enterprise data beyond that which models were trained on. Support layers provide access to model acceleration, optimizing, versioning, caching, inference hosting, capabilities, and more. Vendors like Databricks have latched onto these needs and use the term LLM Ops, but most of these services are common to many AI models — not just language ones.

### Figure 3

**Some Architecture Building Blocks Of GenAI Solutions**








## Use Patterns To Work With AI Development And Operations Teams

Patterns are solution templates created using architecture building blocks. The most common genAI pattern is retrieval-augmented generation (RAG) (see Figure 4). It supports natural language use cases while minimizing or eliminating the need for model tuning, which is time consuming, difficult, and expensive. Within the next year, many of these building blocks will be available as low-code capabilities in platforms from IBM, OpenAI, and Microsoft. Regardless, architects must understand how they work together:

- **An embedding model generates vectors which helps enrich prompts.** Steps 1 to 3 unlock enterprise knowledge by processing documents, splitting them into passages, producing dense vector outputs, and storing them. They use special generative models like BERT to do this and store vectors in special databases or caches. Embeddings are mathematical representations of language as high-dimensional vectors. Leading firms are using domain-tuned embedding models to improve the relevance of enriched prompts, producing better model responses. For example, a vector for the word "platform" in the IT domain will differ from an oil- and gas-tuned model. Therefore, a question about platforms for an IT use case should enrich prompts with different information than one for oil and gas.
- **Users enter prompts that are processed by an intent gate.** In steps A and B, applications preprocess natural language inputs and pass them to a retriever. Intent gates accomplish several functions: 1) intercepting inputs that are inappropriate for the solution (e.g., prompt abuse or hacking attempts), 2) rerouting inputs to automations that don't require generative responses, such as a password reset request, and 3) addition of metadata and shaping text to the inputs to help

the retriever enrich prompts appropriately. For example, Microsoft's Bing Chat examines client input questions, inserts metaprompts, and even does some prompt rewriting. These help retriever code create better prompts, which helps align model outputs with the user's and Microsoft's intent. Architects must ensure that genAI teams consider intent-based governance instructions as part of inputs.

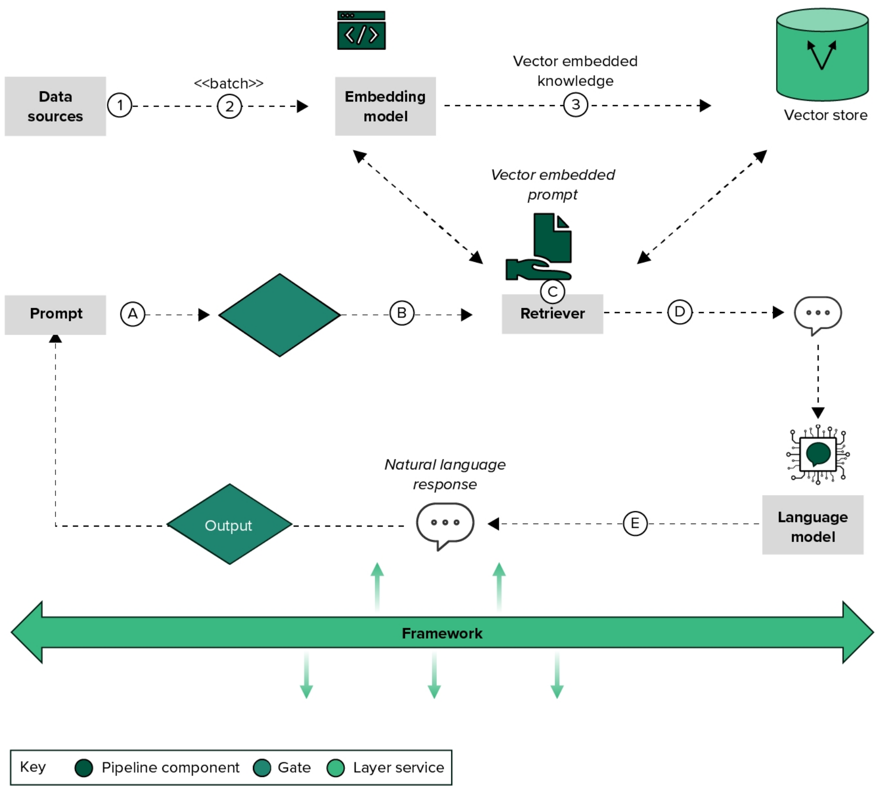
- **A retriever finds and inserts associated knowledge.** The retriever is the workhorse of this pattern. In step 3, retrievers look up associated domain knowledge using vector database search. For example, Meta's original RAG research paper introduced Dense Passage Retrieval (DRP), which uses domain-tuned embedding models to improve search results. LangChain includes an Ensemble Retriever object that combines DRP with traditional semantic search and knowledge graph retrieval methods. Intent-adjusted prompts are enriched with seeds, instructions, and relevant enterprise knowledge, which are sent to a generative model. Architects should help genAI teams experiment with various combinations of data look up and data enrichment to find architectures that increase response helpfulness and harmlessness.

- **A language model and output gate return current, safe, and helpful text.** In step E, a generative model receives the enriched prompt and returns output. These may draw from either the model's trained knowledge, that contained in the enriched prompts, or both depending on seeding instructions. Model responses are then passed through an output gate, which examines generated output for appropriateness, errors, biases, other harmful content, and so forth. Advanced firms like Twilio are building sophisticated trust and safety guardrails that are called from these gates, including checks for input and output consistency (i.e., appropriateness). Architects should help decide for or against language model tuning versus prompt engineering as a path to produce more helpful responses. We find that tuning a generative model is often difficult, expensive, and can yield fewer improvements than just improving embeddings or RAG code.

- **A pipeline framework provides access to support infrastructure, tools, and libraries.** Tools like LangChain or MLflow are specializing in generative model pipeline building. General purpose ML platforms from AWS, Databricks, Dataiku, Microsoft, and others are all introducing genAI-specific capabilities as well. Architects must help acquire and support these tools. They must also help design services like permissioned data access, model caching, versioning, and log file monitoring. Finally, architects must help developers secure public or private access to model training and runtime environments. Work closely with security and risk personnel on these decisions using architecture views.

## Figure 4

**The RAG Solution Pattern For Natural Language GenAI Use Cases**





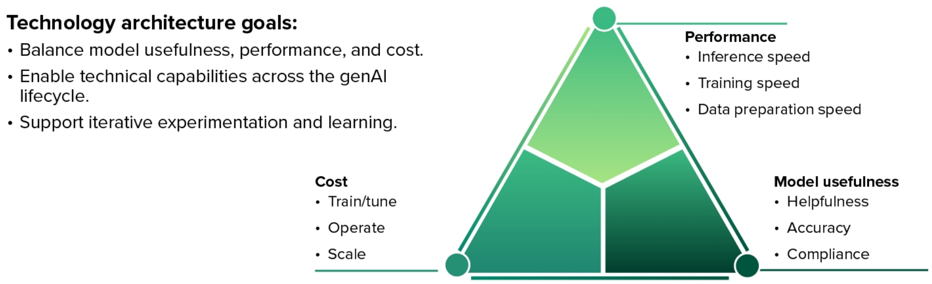

## Create A Technical Architecture That Supports Rapid, Controlled Experimentation

GenAI technology and best practices are evolving at an unprecedented pace — your firm will only learn by doing. Model governance and control as you experiment is crucial given genAI applications' unique characteristics. Neerav Vyas, VP and head of martech and adtech at Tredence, told us, "Innovation cycle time is a crucial metric for genAI success." The faster your firm can iterate ideas, the better off you'll be. When making infrastructure decisions across the AI lifecycle for your engineers, balance alternatives using the three criteria in a generative AI triangle: model usefulness, performance, and cost (see Figure 5). As an architect, you play a big role by:

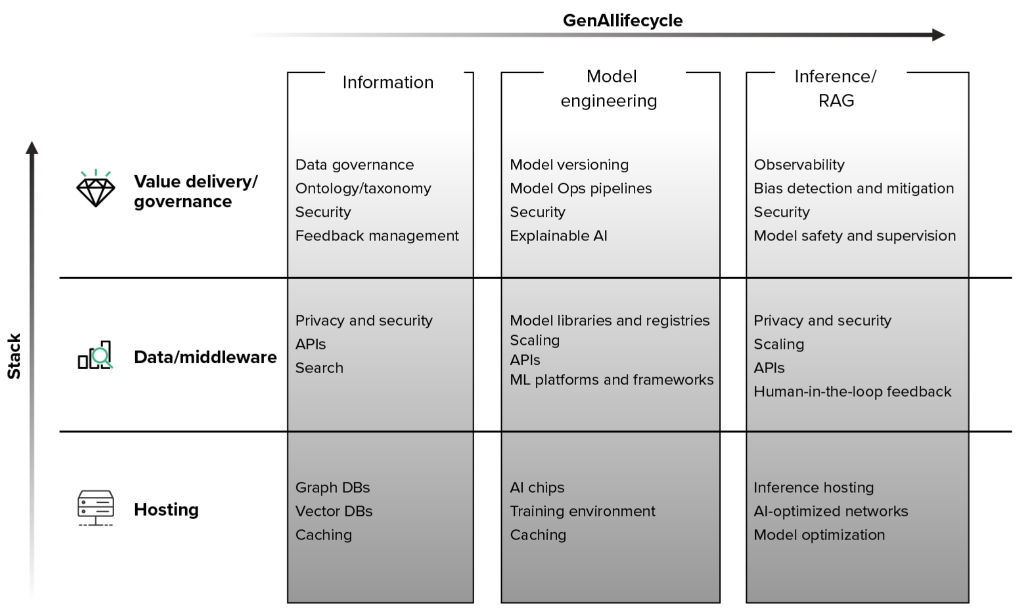
- **Participating in genAI value delivery and governance strategy development.** GenAI patterns are evolving so quickly that architects must help their business anticipate future needs. For example, champion-challenger-style model management requires an architecture that maintains and compares many models, swapping them out as performance changes. As frontier models come to market, model safety and supervision capabilities will be more important. You'll need to understand and make recommendations on features like Anthropic's Constitutional AI or help your business sort through emerging AI regulatory requirements, such as the UK's AI policy and the EU's AI Act. Each of these strategies and more demand your consideration now.
- **Enabling data and middleware integrations.** Almost every software package on the market is adding genAI. Most will need your data, some will offer model tuning and customization, and a few will provide governance and safety controls. Enterprise architects will have their hands full over the next few years figuring out how to supply models with data and integrate genAI tooling and enterprise stacks. For example, ScienceLogic's incident and root cause analysis service is implementing genAI for IT systems, but do you trust the vendor with your data? How will you ensure the responses are correct and lead to timely corrective action? What's your strategy for using Microsoft Copilot's generative capabilities that employ your firm's data?
- **Evolving model hosting infrastructure to be ready for an AI computing future.** Package applications won't create a competitive advantage — if you can buy it, so can your competitors. To support custom application development, architects must inventory their AI hosting capabilities across the genAI model lifecycle. Identify gaps based on the solution patterns your developers need and plan investments accordingly. One data scientist told us that he spent considerable time learning how models worked with various training and inference accelerator hardware options. He also reported facing serious tradeoffs when optimizing model inference performance with a technique called quantization. You need to help with these decisions.

## Figure 5

GenAI Adds A Few Twists To Traditional Technology Infrastructure Management








# Communicate With Governance, Risk, And Compliance Stakeholders

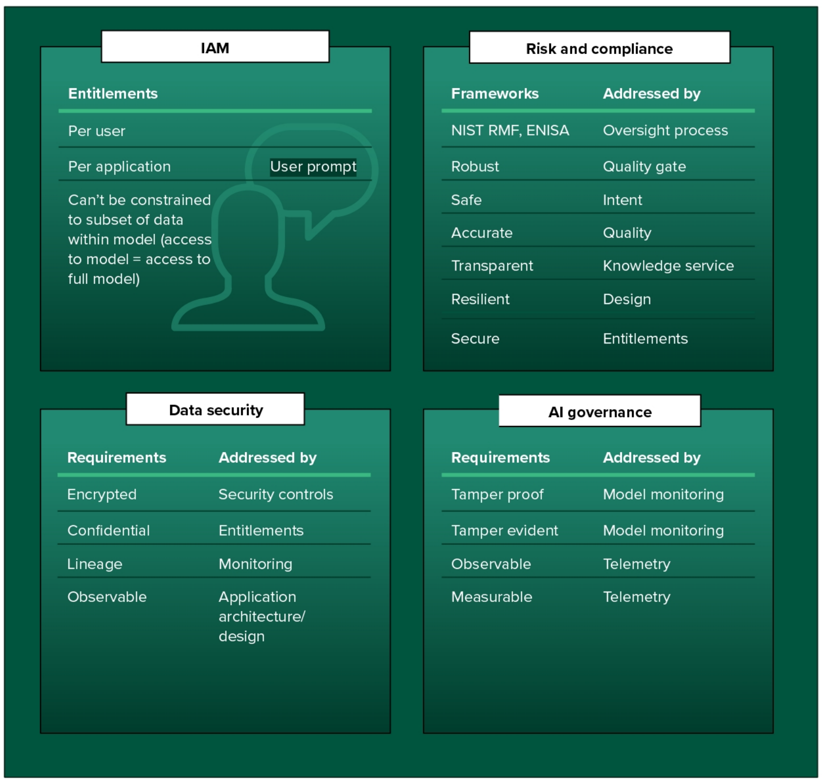
When it comes to genAI, AI decision-makers are most concerned with privacy, data, and misuse. An architecture view is a model that addresses concerns like this with specific stakeholders. Architects should develop views of proposed architectures that address these concerns, and show how various risks will be mitigated to reduce potential harm and increase the possibility of desired benefits (see Figure 6). We recommend you build views for the following stakeholders:

- **IAM stakeholders care about application entitlements and permissions.** Users experience genAI via enterprise and consumer-facing applications. Therefore, existing mechanisms for user and application entitlements and permissioning dictate access to information sent in and returned by prompts. This is your primary layer for data security since there's no mechanism to segment access to data within a model. If an application or user has access to the model, they have access to all data the model was trained on. This requires teams to enforce Zero Trust access principles at a user or application level instead of the data level. This is a big change from other architectures where more granular security options exist.
- **Risk and compliance stakeholders care about regulatory issues right now.** Regulatory, legislative, and partner requirements for genAI continue to pummel GRC teams. There's no end in sight. Each of these external entities uses slightly different verbiage, requirements, and definitions that must be satisfied. Items as basic as ensuring customer data isn't used — or is protected — in model training can be confounding to implement. Bias detection and elimination is another thorny topic; requirements are vague and subjective. Expect to keep copious documentation, possibly a tamper-resistant audit trail, as evidence that will touch multiple areas of your architecture. Helping create these items — and building out a matrix that shows where overlaps between externalities exist — will remove obstacles from teams tasked with oversight by helping them do their jobs.
- **Data security stakeholders care about the entire genAI application lifecycle.** There's no escaping data issues, which fall into three primary domains across the model lifecycle: 1) keeping training data secure, 2) security of data submitted in prompts, and 3) security of data returned via prompts. Each of these items requires a different set of security controls, but the good news is most of those controls already exist today. Sequestration and encryption of training data is widely deployed. Confidential computing techniques mostly address the security of training data. Protecting data sent and returned via prompts is governed by IAM controls, as mentioned above. Plan to use the intent service and quality gates as points of intercept and monitoring to prevent unauthorized data access and leakage.
- **AI governance stakeholders care about model security and integrity.** Model security and integrity is a niche, emerging area with enormous implications. As models proliferate, real attacks designed to tamper with or infer about training data will occur more often. Academic research will come to market soon as better tools for monitoring the integrity of underlying training data and protecting against

inference attacks emerge. Implementing these capabilities will become critical and may span different budget categories from cybersecurity to product teams. You'll need model monitoring, integrity, and AI/ML security controls designed to monitor and protect the underlying data and its architecture. These will require new specialists on both security and ModelOps teams, and possibly new vendors and service partners.

**Figure 6**

**Build Simple Views That Explain How Gates Work To Manage Risk**






# Use Six Architecture Principles To Guide GenAI Decisions, Not Standards

The pace of change makes fixed technology standards a pipe dream. Instead, plan to evaluate solutions against a set of architecture principles. According to The Open Group, architecture principles define the underlying rules and guidelines for the use and deployment of all IT resources and assets across the enterprise. They reflect a level of consensus and form the basis for making future technology management decisions. We categorize principles according to those that help architects evaluate applications and their use of information, and those that help evaluate technology infrastructure and the operational systems they use.

### Information And Application Principles Value Governed Models And Layers Of Intelligence

We recommend three principles for evaluating information and application elements of proposed applications:

- **Principle 1: Use the right model for the right task.** This principle recognizes that genAI-powered applications are never built on a single LLM, but rather require different models to fulfil different tasks. Architects should judge proposed investments by how well or poorly they promote access to (and expertise in using) the right model for the right application task. Seek to grow expertise in model tuning, prompt enrichment, and input/output governance.
- **Principle 2: Include AI in the loop.** This principle guides investments toward the design of closed-loop learning systems that include AI in support of humans, instead of the other way around. Use the robotics quotient (RQ) as a metric for AI collaboration readiness. Gates must revolve around human judgement and should evaluate and filter user intent; they also evaluate outputs against model usefulness and a taxonomy of potential harms. Build systems to capture learnings from both domain experts and users, feeding it back into future tuning, RAG, and gate updates. Finally, as genAI-powered applications are given the power to automate actions, systems must evaluate decision transparency and appropriateness.
- **Principle 3: GenAI models thrive on data.** This principle acknowledges the role of data and insight in genAI solutions. Using this principle, architects must secure funding to invest in robust and evolving layers of intelligence that lead to direct value in genAI-powered applications. While traditional types of data are important (e.g., customer, product, sales, and marketing), content and documentation are also critical for training models. An emerging genAI trend is linked knowledge represented as a graph. Graph databases like TigerGraph or Neo4j are gaining popularity. Advanced firms are loading and linking their source data and content into graph-based knowledge representations. The resulting linked data is an excellent source for model training and augmented retrieval, but also presents cost and performance challenges.

## Technology And Operations Principles Balance Cost, Performance, And Effectiveness

Adopt three principles to help your firm operate with many models, for many tasks, in many use cases:

- **Principle 4: Models can be easily updated and replaced.** This principle recognizes the rapidly evolving state of genAI technology. It guides architects to support investments like containerization of inference endpoints that lower the cost of both upgrading and tuning models or swapping one embedding model for another. Architects should review model pipeline configuration, asking engineers about the cost to make changes. They should also support the development of APIs that help engineers make technology changes as new patterns and tools evolve.
- **Principle 5: The architecture can scale efficiently over orders of magnitude of growth.** Expect growth in all dimensions over orders of magnitude: 10 or 100 times more models will likely require 10, 100, or 1,000 times more data over the next few years. Plan to increase the number of genAI models managed, using tools like model version control from Neptune.ai. It'll be tempting to put all your eggs in one vendor basket; this principle guides you in the opposite direction. Finally, it recognizes the need to scale up for model training and tuning. Understand how various models interact with different AI acceleration chips and provide tools for model compression that trade off model usefulness, operations cost, and performance.
- **Principle 6: Proactively manage the entire lifecycle, including RAG.** Prioritize investments that automate the entire genAI lifecycle, from investment and requirements to development, version control, automation, deployment, and monitoring. Support genAI pipelines with frameworks like MLflow, and favor technical architectures that provide a variety of model hosting options. Facilitate tradeoffs between, for example, on-premises training with owned GPU assets versus access to innovations in the cloud. Architects should support model training and tuning pipelines that efficiently separate business logic from models, separate tools and infrastructure from pipelines via services and APIs, and recognize that prompts are important application assets that must be managed as well.

# Supplemental Material

## Companies We Interviewed For This Report

We would like to thank the individuals from the following companies who generously gave their time during the research for this report.

Boise Cascade

Dan McCreary

Forrester Research (Izola Team)

Kinkaid

LivePerson

Microsoft

Scale AI

Tredence

Twilio

West Monroe

## Analysts Involved

This report represents a cumulation of collective knowledge from many analysts studying genAI. We would like to extend a thank you to everyone who helped that we could not mention above.

Alla Valente

Boris Evelson

Brandon Purcell

Michele Goetz

FORRESTER®

# We help business and technology leaders use customer obsession to accelerate growth.

FORRESTER.COM

## Obsessed With Customer Obsession

At Forrester, customer obsession is at the core of everything we do. We're on your side and by your side to help you become more customer obsessed.

### Research

Accelerate your impact on the market with a proven path to growth.

- Customer and market dynamics
- Curated tools and frameworks
- Objective advice
- Hands-on guidance

Learn more.

### Consulting

Implement modern strategies that align and empower teams.

- In-depth strategic projects
- Webinars, speeches, and workshops
- Custom content

Learn more.

### Events

Develop fresh perspectives, draw inspiration from leaders, and network with peers.

- Thought leadership, frameworks, and models
- One-on-ones with peers and analysts
- In-person and virtual experiences

Learn more.

FOLLOW FORRESTER

## Contact Us

Contact Forrester at www.forrester.com/contactus. For information on hard-copy or electronic reprints, please contact your Account Team or reprints@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
Tel: +1 617-613-6000 | Fax: +1 617-613-5000 | forrester.com