

cloudticity

Getting Ready for Generative AI in Healthcare

Table of Contents

INTRODUCTION 4

GENERATIVE AI IS EVERYWHERE! HOW CAN YOU USE IT IN YOUR HEALTHCARE BUSINESS? 5

- AI Improves the Patient Experience 5
  - Chatbots 5
  - Personalization 5
  - Reducing ER Wait Time 6
- AI Enhances the Productivity of Healthcare Workers 6
  - Preventing Mis-Dosing 6
  - Aiding in Diagnoses 7
  - Reducing the Need for Invasive Surgeries 7
  - Automating Administration 7

NOT SO FAST! CHALLENGES TO ADOPTING GENERATIVE AI FOR HEALTHCARE 8

- Data Roadblocks to Generative AI Adoption 8
  - Collect the Data 9
  - Clean the Data 9
  - Properly Label the Data 10
  - Check the Data for Problems 10
  - Use a Tool 11
  - Be Security Conscious 11

WATCH OUT! SECURITY THREATS TO GENERATIVE AI ADOPTION 12
What are the Top AI-Specific Security Threats?..... 12
Prompt Injection 12
Insecure Output Handling 13
Training Data Poisoning 13
Model Denial of Service (MDOS) 14
Supply Chain Vulnerabilities 14
Insecure Plugin Design 14
Sensitive Information Disclosure 15
Excessive Agency 15
Overreliance 16
Model Theft 16
Top Strategies for Securing Your AI Solution..... 17
Apply Security to Every Layer 18
Use a Policy of Least Privilege 19
Use Multiple Accounts 19
Isolate Network Resources 19
Encrypt Everything 20
Enable Traceability 20
Centralize Findings 20
Keep People Away From Data 21
Automate Security Best Practices 21
Prepare for Security Events 22
Use Multifactor Authentication (MFA) 22

WHY CONSIDER CLOUDTICITY AS A MANAGED SERVICE PROVIDER 23
Three Adoption Patterns for Generative AI in Healthcare 23
1. Pay and Use 24
2. Integrate the Models into Your Native Applications 24
3. Enrich the Models with Your Healthcare Data 25

INTRODUCTION

Unless you've been hiding under a rock, you've doubtless heard about ChatGPT and the amazing recent advances coming out of the field of generative AI. These large language models can now write functioning code, craft poems, even answer exam questions, and many other things, in dozens of languages, almost instantaneously.

Naturally, there's a rush to apply this technology in the healthcare sector, and for good reason. Generative AI in healthcare can help both patients and practitioners. It can increase clinical efficiency, improve the patient experience, and even support better healthcare outcomes. For healthcare businesses, this means keeping up with the market.

Of course, if adopting a new technology were easy, you'd already be using it and wouldn't be reading this eBook. Since it's not easy, we're here to help provide guidance on what you need to know and how to get started.

This eBook will cover the benefits of generative AI in healthcare, enumerate various challenges to incorporating generative AI and how to overcome them, and lay out a framework to help you hit the ground running.

Let's get started!

GENERATIVE AI IS EVERYWHERE! HOW CAN YOU USE IT IN YOUR HEALTHCARE BUSINESS?

How can you use generative AI in your healthcare business today? Here's what other businesses are doing.

AI Improves the Patient Experience

First, let's talk about how the patients interacting with the healthcare system are benefiting from AI.

Chatbots

Chatbots are algorithms that can process and respond to natural language. If you've used a customer service portal at an online store recently, you've probably interacted with a chatbot.

In healthcare, chatbots can be used to answer simple questions, help people find information, schedule appointments, clarify billing information, and escalate serious cases to a human.

Personalization

Generative AI can now personalize responses to suit individuals. A chatbot powered by AI could modulate the tone of its replies based on a patient's emotional state, which it could get from *sentiment analysis*. A chatbot with this information could focus on the most distressed patients first, phrase replies that are more comforting, or quickly route such patients to a human who can help them.

AI can also help with translation. If a doctor needs to communicate in another language with a patient, they can do that with machine translation, meaning a reduced risk of medical errors and a better customer experience.

Reducing ER Wait Times

We've all waited at the emergency room for what feels like too long. But AI is changing that. Thanks to AI, we're seeing billboards pop up across the country that say things like, "ER 10 Minute Wait Time". That's because AI can be used to analyze admissions, discharge and transfer (ADT) data and make recommendations that can shorten ER wait times, like sending less severe patients to the nurse and scaling staffing needs based on the day of the week and time.

AI Enhances the Productivity of Healthcare Workers

Now that we've discussed how generative AI in healthcare benefits patients, let's talk about how it benefits workers.

Preventing Mis-Dosing

Patients don't always take their medications correctly, and doctors don't always get dosing recommendations right. In fact, 7,000-9,000 people die annually in the U.S. from taking their medications incorrectly, but AI could help fix this.

A Nature Medicine study used an AI monitoring system that operated in patients' homes to see how they were using either an inhaler or an insulin injector. It was able to detect when they hadn't followed the appropriate steps. By adding generative AI to this tool, businesses could s end alerts to those patients to correct their procedure.



Aiding in Diagnoses

AI is being used to diagnose ailments sooner, like recognizing breast cancer in early stages in CT scans, which significantly increases the chance of survival. And there are AI solutions that can identify patients who are at high risk of diseases like Alzheimer's, which can help drive preventative care that slows the progression of the disease.

Reducing the Need for Invasive Surgeries

Surgeries are some of the most difficult and complex procedures a human being can perform, and even the best surgeons sometimes slip up and make mistakes. Robotic surgery machines are allowing for far more precision in these operations, reducing the likelihood of catastrophic errors. And robotic surgeons can perform many kinds of procedures through tiny incisions, reducing the damage done to the body and the time required to heal.

Additionally, robotic surgery can be performed from anywhere in the country. Patients wouldn't need to travel to the facility of a particular specialist.

Automating Administration

AI-powered virtual assistants have the potential to automate substantial amounts of the drudgery facing healthcare staff. There are software platforms that can schedule meetings, transcribe and annotate notes, follow up with patients, and handle administrative tasks so that clinicians can focus on healthcare.

Services like Microsoft DAX allow clinicians to draft clinical notes automatically, reducing hours of work to mere seconds and enabling them to spend more time with patients and less time typing up notes.

NOT SO FAST! CHALLENGES TO ADOPTING GENERATIVE AI FOR HEALTHCARE

It's easy to want to jump into generative AI with gusto. There's a lot of pressure to use this cutting-edge technology to stay competitive and relevant and to better serve patients.

But there are two main roadblocks to successfully (and safely!) employing this technology, and you should understand them thoroughly before you proceed.

- There are a lot of steps you have to take even to get the data ready for AI. This is almost always the case, but it's especially true for healthcare data because it's so heterogeneous.
- You must protect against security threats that could expose patient data, or worse, compromise the integrity of information that affects the delivery of healthcare services, (i.e. bad information fed to the AI produces bad recommendations for the clinician).

This section will discuss these challenges and how best to overcome them.

Data Roadblocks to Generative AI Adoption

Getting the data ready for AI is anything but simple. Healthcare data is often siloed across multiple environments, such as legacy applications, cloud environments, or on the edge. It can also be written in a variety of different coding languages, like CCDA, HI7v2, CSV, FHIR, and even proprietary languages. Many EHR vendors store data in proprietary coding languages, for example. To top it off, these systems tend to be incompatible, making it difficult to get the data into one central data store where it can be fed to AI.

The data produced by the healthcare industry is also usually unstructured data, such as clinical notes, discharge summaries, lab reports, imaging data from MRI scans, and so on. You generally can't just give unstructured data to an algorithm and expect it to do anything useful, which is a shame because it holds some of the greatest value for healthcare organizations.

That's why proper data preparation and data management are key to being successful with generative AI in healthcare. Here are some things you can do to get your data ready for generative AI.

Collect the Data

First, you have to have data for your model to use, beginning with data collection. We've already noted how variegated healthcare data can be, and this means collecting it will probably be difficult. The best advice we can give you is to start your data harvesting efforts early, and to standardize everything as much as possible. Start with standard operating procedures and training for all staff who code data into the system.

You should also bear in mind that data is increasing at a staggering rate, so there could be far more data in a year than there is today, so plan ahead and make sure that you have adequate storage for all the data you want to process.

Clean the Data

People associate "machine learning" with exciting buzzwords like "neural networks" and coders in hoodies training advanced models late into the night. This is true as far as it goes, but a lot of machine learning boils down to slightly less glamorous work like data cleaning.

Machine learning models are only as good as the data they're trained on, so you must comb through your data to ensure it's as high-quality as possible. This can involve translating date formats and data types for consistency, filling in missing information, dropping anomalies, and much more.

Properly Label the Data

With a few exceptions like certain types of clustering, most artificial intelligence applications require labeled data. These labels are crucial to the process of learning statistical patterns in the data, and without them the models aren't much use.

Labeling data is more or less exactly what it sounds like. If you're training an image classifier that identifies which radiological scans contain tumors, for example, you'll need images with cancer and others without cancer, and they'll have to have labels like "cancer" and "no cancer".

Be aware that data labeling on its own can be an extremely labor- and time-intensive process, so leave plenty of time for it if you're going to create bespoke data for a model.

Check the Data for Problems

Once you've collected, cleaned, and labeled the data you'll need to spend some time exploring it. Appropriately enough, this is known as "exploratory data analysis" (EDA), and it's an important part of ensuring that the data is accurate.

A good way to start EDA is just to generate a bunch of summary statistics related to different parts of the data. You might look at averages for values like age or weight, for example, and then dive into specific metrics if anything looks amiss.

It's also common to generate plots for the same purpose. Bar charts, line plots, histograms, and the like are all great ways of painting a picture with the numbers that are easier to understand and evaluate.

Use a Tool

One strategy that more and more organizations are turning to is the use of dedicated data preparation offerings that simplify and streamline much of the work discussed in the previous few sections.

For data preparation, Amazon SageMaker Data Wrangler is a good choice. It facilitates exploration and visualization, can help identify anomalies, and supports many common data transformations. For extract, transform, and load, AWS Glue can help you discover, prepare, move, and integrate data from multiple sources to prepare it for analytics and AI.

Be Security Conscious

No discussion of data preparation would be complete without a mention of data security, particularly in the context of healthcare; but the next two sections are devoted entirely to this topic, so read on to learn more.



WATCH OUT! SECURITY THREATS TO GENERATIVE AI ADOPTION

We've already covered how Large Language Models (LLMs) and generative AI more broadly have the power to transform the healthcare sector. But these complex, inscrutable algorithms also bring a host of security threats that need to be considered and addressed to protect the privacy of patients. But, it's also important not to neglect the integrity of the information produced by these algorithms, especially when they are driving critical healthcare decisions.

Here are the top 10 AI and machine learning security threats for healthcare, and how to protect against them.

What are the Top AI-Specific Security Threats?

Prompt Injection

One of the most common security threats around generative AI is prompt injection. Recall that a prompt is the basic interface for using an LLM. You'll ask it to generate a working prototype of an application, for example, and it'll generate an output. A prompt injection occurs when a bad actor artfully crafts a prompt to extract information that they're not supposed to have. In healthcare, this can mean personal identifiable information (PII) getting in the wrong hands.

There are two basic steps you can take to guard against prompt injection attacks. First, ensure the models have end-to-end encryption; not only does this make it harder to manipulate a prompt in the first place, it also means that an attacker will have to work harder to decrypt any information that does come out. Second, ensure that developers thoroughly test various prompts to see what response they get and where they need to rectify any vulnerabilities.

Insecure Output Handling

Insecure output handling occurs when data fed to or generated by the model isn't effectively validated. When building generative AI applications, there must be layers of data verification and quality control

One way to approach this task is to ensure that access to the LLM is as limited as possible – data engineers, for example, might only be granted access to a single tier of data, rather than allowed to pore over everything you have. Another solution is manually testing your model. Testers can prompt a model with myriad questions and inputs to see if they can cause the model to violate security best practices. If they're successful, you must vet your protocols before exposing the model to end-users.

Training Data Poisoning

Artificial intelligence is not magic. For all the obvious power of LLMs and generative AI, these tools are only as good as the data they're trained on. For this reason, there are a number of ways data can be tampered with or otherwise compromised, including what's known as "training data poisoning". This involves corrupting the model's training data so that it makes an incorrect recommendation when it sees certain healthcare information.

Just as the development of an infant can be stunted by teaching it the wrong words for the objects in its environment, models in the development and training stages are vulnerable to misinformation. If not caught and rectified, this can lead to inaccuracies and biases down the line.

In medicine, it's often said that prevention is key. The same applies to data poisoning. As the volume of data LLMs are trained on increases, it will become steadily more challenging to weed out incorrect or malicious information retroactively. If you discover any questionable data, go back into the data archives, track down the data uploads for each time that you trained your model, and comb through to determine what needs to be flagged and removed. Some parts of this can be automated.

Model Denial of Service (MDOS)

A denial of service attack refers to any attempt to take down a machine, application, website, etc. by spamming it with traffic or requests until it collapses. As the name implies, a model denial of service (MDOS) attack is the AI version of this villainy. Because model endpoints are often publicly available, they can be relentlessly spammed, thus making them unusable.

Patients and providers have critical moments when they need immediate access to healthcare resources or information; when this information isn't easily accessible, the entire healthcare process is slowed down. A model denial of service means that legitimate actors cannot access a generative AI tool when it's needed.

Fully protecting your application from an MDOS attack could be a whole other eBook entirely, but the simplest way to guard against this potentiality is by putting model endpoints inside of a virtual private cloud (VPC), where you can control and limit access to it.

Supply Chain Vulnerabilities

Whether we're talking about a factory or an entire suite of healthcare-focused LLMs, the supply chain has many different moving parts. This means many different points at which a bad actor can embed themselves, including hiding within plugins, embedding malicious software, or tampering with training data. Insecure plugin design is one of the primary issues leading to vulnerabilities in the supply chain, often resulting from not having robust encryption protocols that ensure HIPAA-protected data isn't disclosed.

To secure the generative AI supply chain, end-to-end encryption is simply non-negotiable.

Insecure Plugin Design

Having raised the issue of insecure plug-ins, let's say a little more about them. Plug-ins can be the one thing that makes or breaks an effective LLM. And, since they're frequently embedded all across the supply chain, any weak link can create an access point for a bad actor. There's no alternative to carefully inspecting every plugin at every stage, from programming to maintenance, to ensure that vulnerabilities are addressed.

Sensitive Information Disclosure

Have you ever gotten an email that was intended for someone else, or overheard a conversation you weren't supposed to hear?

When it comes to LLMs, this is a significant threat that occurs when a model is not properly trained and programmed.

There are a few different kinds of sensitive information disclosure – a healthcare provider might accidentally be shown the wrong person's data, or they might receive too much of the right patient's personal information.

To prevent this, you must be very careful about matching models to end-user personas using role-based access. What's more, there needs to be a comprehensive way to compartmentalize information, offering it up on a strictly need-to-know basis.

Excessive Agency

You've no doubt had colleagues who overstepped their authority or had friends who tried to offer advice when they were neither qualified nor experienced enough in a given area to do so (the fancy word for this is "ultracrepidarianism").

Well, "excessive agency" is a similar concept, referring to situations in which people use LLMs beyond their appropriate scope. This can lead to a model being exposed to information it's not supposed to see, which in turn could lead to data breaches or other problems.

Preventing this starts by ensuring all security systems are aligned across all elements that touch the LLM. You need to have robust policies detailing which data your models can access and which data is off limits, and these need to be understood by everyone using generative AI in your organization. You should also automate these policies.

Overreliance

While LLMs can provide tremendous value and insights, it remains necessary to keep a human in the loop to check that returned data is correct. End users might not always be checking the model's output, and if no one else does either, that will eventually be problematic.

This necessity stems from basic facts about the way LLMs function. At the end of the day, they're enormous statistical machines able to map inputs to outputs. Though they're often surprisingly capable, they have no built-in ability to wonder about their own sources or the veracity of their own conclusions (like many humans do).

In particular, LLMs are well known to "hallucinate", i.e. "make things up out of whole cloth". You've no doubt already seen examples of people asking ChatGPT for a list of publications by a well-known scientist, only to get back a mix of accurate information and complete fabrications.

We probably don't need to point out how bad it would be if a model hallucinated illnesses, or invented non-existent patients, so be sure to have someone responsible for fact-checking your generative AI!

Model Theft

Just like a soft drink company will go to great lengths to protect the secret sauce that makes its drinks so popular, you have to strike a balance between making a model that's hard to steal but easy to use.

Encryption, obfuscation (i.e. hiding or concealing important elements of the code), and paying a close eye to any irregularities or unwanted visitors are your best bets.

When developing an LLM, take some time to think in the shoes of a bad actor, and test out various data exfiltration methods to see how easy it is to steal or reverse engineer the LLM. This is sometimes known as developing a "security mindset", and it's an enormous asset.

TOP STRATEGIES FOR SECURING YOUR AI SOLUTION

While generative AI has the power to transform countless industries like healthcare, it also raises a bevy of security concerns.

Despite the (largely justified) excitement and buzz surrounding LLMs, developers need to keep in mind the fact that bad actors are seeking to exploit the technology to spread misinformation, steal valuable patient data, or even abscond with your valuable intellectual property.

With this in mind, here are the top ten most impactful security best practices for security your AI solution.



Apply Security to Every Layer

The first best practice applies to basically any situation, and it's the injunction to utilize sound security protocols for all the layers of your project, including:

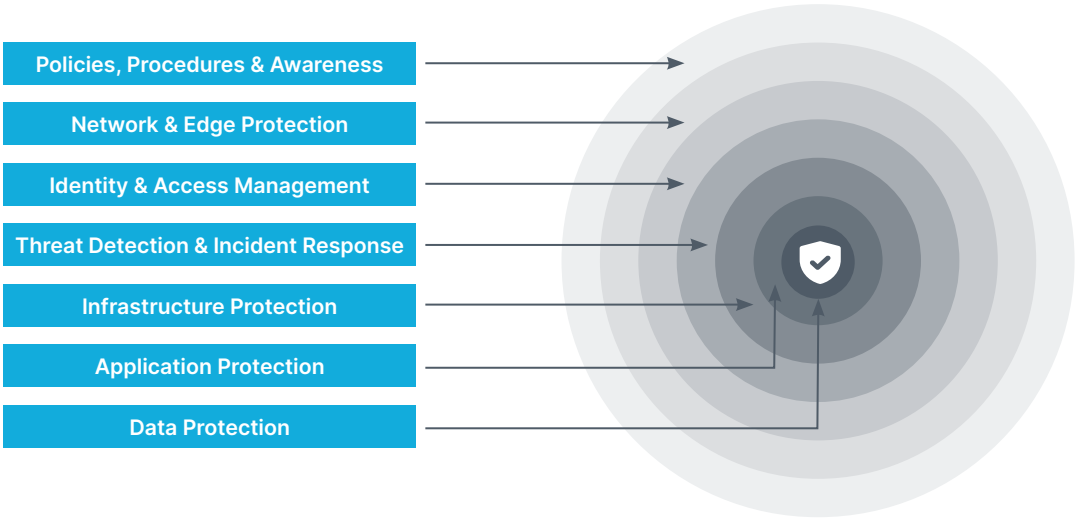
- Data Protection: We're trying to protect the data at the lowest possible level. We want to employ security at every layer above that to detect security incidents and respond quickly and effectively. For data protection, we want to ensure that data is encrypted in transit and at rest and employ least-privilege access policies on resources that contain sensitive data.
- Application Protection: The next layer is the application layer, which addresses the applications communicating with the data. This involves ensuring that the people who are accessing that information have the right user and access permissions.
- Infrastructure Protection: In an on premise data center this means securing your perimeter and access to servers. If you're using a public cloud provider, your vendor will handle much of the infrastructure security, but you still have to manage how that gets deployed and how you allow access to those maintaining it.
- Threat Detection and Incident Response: You have to ensure you have monitoring in place to catch any threat that comes your way and respond to it promptly.
- Identity and Access Management: How do you allow appropriate level access to people accessing company resources?
- Network and Edge Protection: When and how do you allow third parties to communicate with your infrastructure and our network?
- Policies, Procedures, and Awareness: How you understand and create guardrails that restrict how application access is done.

With generative AI, you'll also want to broaden your concerns to include any data being fed to the model and any output generated by the model, as either could contain PII. This is very important, because once a model has seen PII, it becomes all but impossible to ensure that this PII doesn't show up in a later response.

HOW TO ACCELERATE AI & ML INNOVATION WITH SECURITY & PRIVACY

Defence-in-Depth

IDENTIFY, PROTECT, DETECT, RESPOND & RECOVER



Use a Policy of Least Privilege

In information security, the "principle of least privilege" refers to the idea that a person, application, or entity should only be given access to the specific data they need to do whatever work they need to get done. Instituting this as a policy – i.e. as a "policy of least privilege" – does a lot to improve security posture, as it means that your attack surface is dramatically reduced and it's much harder for ransomware or malware to spread throughout your organization.

Use Multiple Accounts

If using a public cloud service provider, which is what we recommend, use multiple accounts for authentication purposes. No one should have access to everything, as this introduces a single point of failure that could compromise your entire technology stack if breached.

Isolate Network Resources

You should also isolate your network resources, for the same reason that you should use multiple accounts. This means preventing (or otherwise carefully controlling) your model from being accessed from the general internet.

If you've spent any time on the internet, you know that some of those using it aren't exactly upstanding model citizens. When they get access to any new technology, including your generative AI offering, they'll attempt to jailbreak or otherwise misuse it. This is a misstep you can't afford when working with sensitive healthcare data.

Encrypt Everything

Another fairly basic modification you can make to improve your security posture dramatically is to encrypt everything – the data the model is trained on, any messages being sent to it, any text it's generating, etc.

Encrypt at rest and encrypt in transit. One subtlety worth mentioning is that you must ensure that the encryption standards used across the stack – between components and plugins, between users and the model, etc. – should be consistent. Without consistency, you could get in a situation where a dev environment has PHI in it. This may sound fine, but the dev team is likely to spin up instances that they may or may not remember to shut down or fully protect. This creates an easy place for an attack. To combat this, encrypt sensitive data all the time, or use synthetic data in development.

Enable Traceability

Traceability refers to your ability to tell who is interacting with your application, at what point they entered (i.e. whether they hit the API or accessed your infrastructure directly at some point), and what they did while they were there. This will facilitate monitoring any changes to the model or its scaffolding, and make it easier to identify any potential attacks or problems.

Centralize Findings

If you've enabled traceability, used a policy of least privilege, or taken any of the previous steps, you should now have a monitoring system that generates information related to what's happening inside your application. Whenever possible, bring all this data together in one place. This will allow you to quickly assess activity occurring along any attack vector; you'll be able to understand whether an attack consisted of one event or multiple related events, whether it originated with one account or several, etc.

All of this is crucial for mitigating damage and closing off those weaknesses in the future, but by default, the relevant data tends to be scattered across different logging tools. One advantage of implementing a security lake is that you can centralize these findings, but, as is usually the case, there are other ways to skin this particular cat.

Keep People Away From Data

You may have heard the phrase "information wants to be free", which refers to a general philosophical openness for data. Where security is concerned, however, this won't do because some people want to use data for nefarious ends – and this is especially true of valuable healthcare data.

Once you've centralized your findings following the previous best practice, you must be very careful about who will see those findings. Not only does excessive openness mean a great threat surface, it also makes it harder to find and patch vulnerabilities if you do experience some kind of breach.

You should only grant data access to people who really need it, and even then, they should only get access to what they need, when they need it, to do their jobs correctly.

Automate Security Best Practices

Even with the best monitoring and alerts, you'll still need security automation in place. This means, for example, setting up systems that automatically implement network protections and policies when someone is deploying something new in your environment.

You also need to automate incident response. You could have alerts going off, for example, telling you that someone is inside of your network trying to get access to your data. Still, if you don't have a way to respond and contain that type of attack automatically then you're at the mercy of employees, hoping someone will see the alert and respond correctly before the attacker has time to exfiltrate any data.

Prepare for Security Events

Finally, you must create a regular cadence for testing your systems and teams under different security scenarios. You should be checking to make sure your automation holds up, that the relevant people are responding correctly, and, in general, the things you expect to happen are actually happening.

All the best practices in the world won't help you if you're caught totally unprepared when a threat emerges, so you need to be consistently testing to make sure you're ready if it does.

Use Multifactor Authentication (MFA)

MFA is one of the easiest ways to ensure that sensitive data are being accessed by appropriate individuals. It's usually not hard to create and enforce a policy of using MFA across your infrastructure, and you should be doing this to protect yourself as much as possible when using generative AI..

Your MFA policies should extend not only to people within your team, but also to customers, patients, and partners. This is one of the best ways of preventing malicious actors from infiltrating your systems by pretending to be the people you're serving.

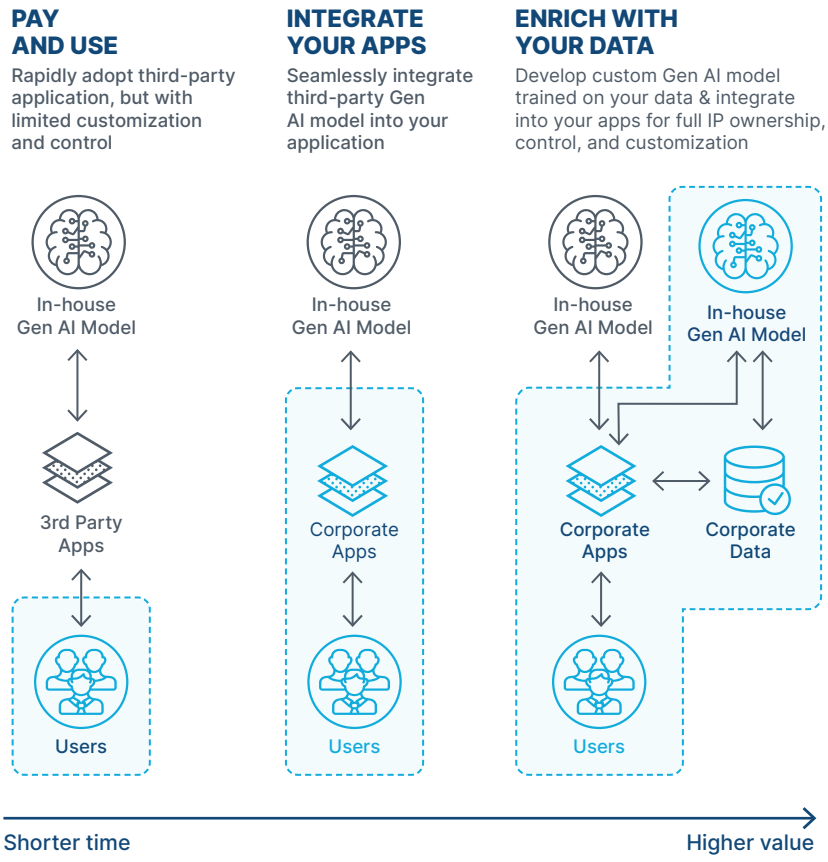


HOW TO GET STARTED WITH GENERATIVE AI

By now, we've covered what generative AI is, how you can use it in your healthcare business, some of the unique challenges facing this technology, and how to address them.

With that monumental work done, let's now turn our attention to how to begin incorporating generative AI into your healthcare solutions. To do this, we'll walk through common implementation patterns different organizations are using.

THREE ADOPTION PATTERNS FOR GENERATIVE AI IN HEALTHCARE



1. Pay and Use

The pay and use strategy is common among teams who either don't have the time or the expertise to become experts in the underlying technology themselves. In such a case, they opt instead to pay to use a 3rd parties platform. Today, popular generative AI platforms include OpenAI and Anthropic, but there are more springing up all the time.

The advantage of this approach is that it lets you hit the ground running in very short order. The disadvantage, of course, is that you're relying on someone else for a critical business function. This means if their models suddenly start to decline in performance – a phenomenon known as model degradation or model decay – there's little you'll be able to do about it.

On the whole, the pay and use strategy is best suited for drafting code and marketing copy, but given the fraught nature of healthcare data, we don't advise it for companies in the healthcare space.

2. Integrate the Models into Your Native Applications

Once you've picked a model vendor and are ready to go, a sensible next step is to start exploring how to integrate generative AI into your existing applications. There are so many ways to do this that it's not possible to exhaustively list them all, but we can provide a few ideas as initial jumping-off points.

An obvious place to plug in a tool like ChatGPT is in your customer service portal, where it can help answer simple questions or route patients to articles that'll help them. Beyond this, it can also be used to train new hires in your internal processes or form part of a virtual assistant that raises the productivity of your workforce.

One thing to be aware of is that model vendors differ in a few ways that will impact how your integration efforts fare. Different models have different "context lengths", meaning that they can process different amounts of information. Anthropic's Claude has a much longer context window than ChatGPT, for example, so it can process longer pieces of text in one go.

In addition, each vendor will have a different way of structuring their API and charging you for using their model. Be sure to read the fine print about token limits and so forth so you don't end up paying more than you expected to.

There are a number of cloud-native technologies you can utilize for the integration strategy. Azure OpenAI Service and Amazon Bedrock make it easy to leverage foundation models from Anthropic, OpenAI, and other companies. You'll be able to use standard REST APIs for languages like Python, Java, etc., making integration a snap.

3. Enrich the Models with Your Healthcare Data

Large language models are very capable out of the box, but there's a lot you can do to make them much more capable.

One example comes in the form of a technique known as "retrieval augmented generation", in which you furnish the model with information sources or context in which it can ground its replies.

This method works well for making AI more effective, but by far the more aggressive option is to fine-tune the model on your specific data. Fine-tuning actually changes the internal structure of the models, making them much better at performing tasks related to your business. If you want to bias a model toward being much better at generating healthcare-related content, for example, then it helps to show it a lot of such content.

Remember that these models are just statistical artifacts. An out-of-the-box model that hasn't seen the specialized terminology that crops up in healthcare won't be as good at parsing it, but you can change that through fine-tuning.

Of course, this brings challenges. Retrieval-augmented generation is usually not that hard to set up from a technical perspective, but the same cannot be said for fine-tuning. If you want to fine-tune a model, you'll need some fairly high-level engineering chops as well as plenty of fine-tuning data for the model to pursue.

As with the integration strategy, there's plenty of cloud-native tools available with functionality for an enrichment approach. Amazon SageMaker, Google Vertex AI, Azure Machine Learning, HuggingFace Hub, and NVIDIA NeMo are the most popular services for training and fine-tuning models on your data. HuggingFace Hub also provides a wide range of open-source Foundational Models.

There are a few things you should be aware of, however. First, data quality matters a great deal in fine-tuning, and you'd be better served curating a small but very good dataset as opposed to throwing everything you have at the model and hoping for the best. Second, once a baseline model has been fine-tuned its performance in other domains may be compromised, so consider that as you plan out your model suite. Finally, fine-tuning is nowhere near as expensive as training a model from scratch, but it's not cheap either. Be sure to keep an eye on those compute costs!



GETTING STARTED WITH GENERATIVE AI IN HEALTHCARE TODAY

One thing is for certain, the race is on. Businesses across all sectors are now keenly exploring Generative AI solutions, and how they can use them to provide the most value. Healthcare is no exception.

However, the journey to integrate generative AI is more of a marathon than a sprint. Despite this, business leaders must engage with it urgently, or risk falling behind the market.

Are you ready to create generative AI solutions for healthcare? Cloudticity and SoftServe might be able to help. We help you get your data ready, design and deploy a solution, and then manage and optimize that solution.

Reach out for a free consultation today to learn more.


cloudticity
Meet with an healthcare cloud expert today!
SCHEDULE FREE CONSULTATION