TECHNICAL WHITE PAPER
May 2024

VMware Private AI – Privacy and Security Best Practices

vmware®
by Broadcom

Contents

Private AI in the enterprise data center – empowering control and security **4**
The principle of shared responsibility in Private AI **6**
Securing the future – threat modeling for Gen-AI applications **7**
VCF platform **8**
The CIA triad – guiding model in information security **10**
Confidentiality 10
Securing model access 10
User privacy controls 11
Platform controls 12
Data encryption 12
Access control 13
Model security 14
Integrity 14
Availability 16
Air-gapped environment **17**
Securing the Gen-AI application **19**
Full end-to-end workflow of a RAG architecture 19
Building the foundation for retrieval: the indexing process in RAG architectures 20
Indexing process 20
Loading diverse data sources 20
Transforming data for efficient processing 21
Tokenization – the building blocks of meaning 21
Embedding – capturing semantic meaning 21
The stored foundation 21
Data preparation and security 21
Retrieval – efficiently finding relevant information 22
Retrieval process 23
Understanding user queries 23
Matching queries with encoded information 24

Prioritizing relevant passages 24
Preparing information for the LLM 24
Feeding the LLM 24
Going from generation to user experience – the final steps in RAG 25
Polishing the response – post-processing 25
Tailoring the response for presentation 25
Seamless integration – user interface and presentation 25
User presentation and interaction 25
Maintaining a positive user experience 26
Private AI network security 26
Securing Private AI ingress traffic 26
Securing Private AI ingress traffic 28
Securing egress traffic from Private AI workloads 28
VMware vDefend 29
Using VMware Firewall to protect Private AI deployments 31
Conclusion 34
References 35
About the author 35
Acknowledgments 35

Private AI in the enterprise data center – empowering control and security

In today's data-driven world, the sensitivity and privacy of information are important concerns for businesses. Traditional AI solutions frequently rely on sending data to external servers, heightening anxieties surrounding data privacy and security. Particularly in light of protective regulations like the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and the Gramm-Leach-Bliley Act (GLBA).

Dependence on external AI-related APIs further amplifies insecurity, leaving businesses vulnerable to price fluctuations, limited control, and feature deprecation. Additionally, the unpredictable behavior of proprietary SaaS models, which are susceptible to unannounced updates, jeopardizes performance and compliance.

Private AI is an architectural approach that aims to balance the business gains from AI with the practical privacy and compliance needs of the organization. Private AI allows organizations to bring AI and ML technologies in-house, reclaiming control over proprietary data, models, and intellectual property. It provides the tools and flexibility to deploy and manage Private AI workloads securely, privately, and efficiently, enabling innovation and success driven by responsible AI practices. Private AI allows organizations to:

- **Innovate with confidence:** Experiment and refine AI models quickly on internal data, free from external dependencies.
- **Tailor solutions to their specific needs:** Craft AI models that perfectly fit their unique data sets and business requirements for superior performance.
- **Stay ahead of the curve:** Quickly integrate and fine-tune the latest AI models, ensuring they're always at the forefront of technological advancements.
- **Build trust and transparency:** Build trust with stakeholders, especially in data-sensitive industries, with on-premises data, improved explainability in AI models, and controlled data lineage.
- **Simplify compliance:** Local data storage and customizable security within your on-premises environment make compliance with data privacy regulations easier.
- **Boost efficiency:** Understand exactly how resources are used for AI tasks to reduce expenses.
- **Control:** Control over the model provides greater efficiency and performance while reducing a customer's supply chain risk by eliminating lock-in to a single vendor.
- **Develop internal expertise:** Build a team of AI specialists within the organization, fostering knowledge sharing and safeguarding proprietary models.

Private AI offers a path to ultimate choice, granting unrestricted control over data and technology. Businesses can keep their proprietary data secure within their controlled environments, ensuring compliance with data privacy regulations. Hosted models allow for complete control over versions, updates, and performance optimization to reduce supply chain risk, eliminating disruptions caused by unexpected external changes. Moreover, Private AI empowers organizations to maintain autonomy and freedom from external vendors' pricing and terms, avoiding vendor lock-in.

The Gen-AI landscape is a fast-moving ecosystem with rapidly changing large language models (LLMs), data science toolkits, and frameworks. VMware Cloud Foundation (VCF) offers a stable infrastructure platform that supports an open ecosystem of AI partners. VMware by Broadcom takes it a step further by providing an architectural approach that balances the advantages of AI with the security and compliance needs of the organization. In 2023, VMware published reference architectures with key partners including NVIDIA, IBM Watson X, Intel, Hugging Face, and AnyScale.

The benefits extend far beyond security. Private AI enables businesses to take control of their AI expenses, avoiding unexpected third-party fees. By understanding precisely how resources are used for AI tasks, organizations can make informed decisions and optimize their IT budgets. Additionally, minimizing reliance on external vendors leads to potential cost savings and greater control over IT budgets.

Private AI fosters agility and innovation by enabling businesses to quickly test and refine AI models on internal data without external service wait times. Customization of AI models to specific business needs and unique data sets leads to improved performance and accuracy. Furthermore, organizations can stay ahead of the technology curve by integrating and fine-tuning the latest AI models as they become available.

Collaboration and knowledge sharing within the organization are also significantly enhanced by Private AI. By securely sharing and collaborating on AI models, organizations advance innovation and knowledge sharing. Additionally, by managing and maintaining models built in-house, AI expertise safeguards sensitive AI models and algorithms from unauthorized access or replication. Moreover, developing in-house AI expertise is crucial for organizations seeking to fully leverage AI and gain a competitive edge. This requires teams that possess a deep understanding of the industry, data, and unique business requirements.

Trust is another key advantage of Private AI, especially for industries with stringent regulations or sensitive data. Organizations employing Private AI build trust with stakeholders by ensuring data privacy and a degree of explainability. Retrieval Augmented Generation (RAG) architectures should include source citations, giving the organization direct control over the components used in the Gen-AI applications, including the datasets used to fine-tune the neural network models and data sources for retrieval.

Additionally, Private AI mitigates the risks associated with moving large datasets to the cloud, such as data gravity. Returning datasets on-premises can be very expensive and time-consuming. Large in-cloud datasets can create vendor lock-in, limiting the organization's flexibility and potentially increasing costs over time. Furthermore, Private AI reduces data privacy concerns: moving datasets, even if they are anonymized, to the cloud increases the risks of the data being used for purposes beyond your initial agreement, such as training third-party AI models.

Furthermore, on-premises data processing simplifies compliance with data privacy regulations. Organizations may employ enhanced customization by combining proprietary datasets with fine-tuned models for superior performance on specific tasks.

The message is clear: Private AI allows organizations to leverage AI while addressing crucial data privacy, security, and control concerns. By carefully considering the benefits and challenges, organizations can determine if Private AI is the right path for their AI workloads. It's time to seize power and pave the way for a future of innovation and success.

The principle of shared responsibility in Private AI

The principle of shared responsibility highlights that security and privacy with Gen-AI workloads is a two-way street. The infrastructure and operations (IT Ops) team ensures the security of the platform and the infrastructure services, avoiding malicious intent to reach or infiltrate the Gen-AI application. The data science team is responsible for securing the application and its processes and ensuring the privacy of the generated information and used data, while safeguarding the infrastructure from malicious intent.

When it comes to Private AI, understanding the difference between data privacy and security is crucial. While they're connected, they are not the same. Let's break down what each means and how everyone can work together to keep the organization's information safe.

In Private AI, privacy focuses on how the organization's data is used. Privacy guarantees the confidential and ethical handling of inputs, interactions, and outputs. Security protects the data from unauthorized access or breaches; it prevents data manipulation, misuse of data, and denial of service attacks.

The CIA triad, a cornerstone of information security, represents three critical aspects: confidentiality, integrity, and availability. In the context of AI-enabled workloads, these principles directly impact both privacy and security. Let's explore how the IT Ops and data science teams are involved in this balancing act.

Confidentiality: Safeguarding sensitive data used to fine tune and operate AI models. This data might include personal information, sensitive organizational data, or any other information that could be misused if exposed. Data scientists, responsible for model selection and development, need access to this data while ensuring it remains confidential. IT Ops is responsible for implementing security measures like access controls and data encryption crucial to prevent unauthorized access.

Integrity: Maintaining the accuracy and trustworthiness of the data and the AI models themselves. Manipulation or corruption of data could lead to biased or inaccurate results. A data scientist is particularly concerned with data integrity. IT Ops and security analysts play a vital role in implementing safeguards like intrusion detection and code verification to maintain the integrity of AI systems.

Availability: Ensuring that authorized users have access to the AI-enabled application and its output when needed. Downtime due to cyberattacks or technical glitches disrupts critical operations. DevOps engineers and IT Ops ensure the infrastructure supporting AI workloads is robust and can withstand disruptions.

Now that we have a better understanding of the personas involved and how the CIA triad underpins AI security, let's explore how to translate these principles into actionable measures. Here, threat modeling frameworks like STRIDE come into play. STRIDE provides a systematic approach to identify vulnerabilities by considering different attack vectors. By mapping these potential threats onto the CIA triad, we can pinpoint weaknesses and develop concrete security controls to safeguard the confidentiality, integrity, and availability of the AI workloads.

Securing the future – threat modeling for Gen-AI applications

Just as traditional applications and systems have vulnerabilities, Gen-AI applications introduce unique attack vectors that require careful consideration. To navigate this evolving landscape, proactive security measures are essential. There are several threat modeling (TM) methods available; the most mature TM framework is STRIDE. Organizations of all sizes can adapt STRIDE, a versatile tool that focuses on fundamental security concerns, to any domain where information security is important.

The STRIDE methodology is a powerful tool for identifying and mitigating threats associated with Gen-AI applications. Independent agencies such as the European Union Agency for Cybersecurity identify STRIDE as a starting point for AI threat modeling. By understanding the potential pitfalls and using a structured approach to threat modeling, the data science and infrastructure teams can build a more secure foundation for the organization's Gen-AI applications. STRIDE is an acronym for six different security threats, as described in table 1.

Table 1. STRIDE – spoofing, tampering with data, repudiation, information disclosure, denial of service, elevation of privilege

Threat	Description	Desired security property
Spoofing	A user takes on the identity of another. For example, an attacker takes on the identity of a system administrator.	Authentication
Tampering with data	Information in the system is modified by an attacker.	Integrity
Repudiation	Information about a transaction is deleted in order to deny that it ever took place.	Non-repudiation
Information disclosure	Sensitive information is stolen and sold for profit.	Confidentiality
Denial of service	Exhausting resources required to offer services.	Availability
Elevation of privilege	Similar to spoofing, but instead of taking the ID of another, the attacker elevates their own security level to that of an administrator.	Authorization

Spoofing poses a threat when an unauthorized individual pretends to be another user or device to access a system. Spoofing poses a significant threat to the integrity of Gen-AI applications. Attackers might impersonate legitimate data sources, feeding the underlying data system with fabricated information during the indexing stage. LLMs may then generate misleading or inaccurate information based on this false data. Additionally, malicious actors could attempt user impersonation to gain unauthorized access and potentially manipulate the system's functionalities.

Tampering threats revolve around unauthorized modification of data, machine learning models, or system components within the Gen-AI application. Tampering could involve modifying fine-tuned data to bias model outputs, altering model parameters to generate incorrect results, or manipulating system components to undermine the integrity of the Gen-AI application.

Repudiation threats refer to situations where attackers can perform malicious actions within the Gen-AI application without leaving behind evidence of their activities. Attackers could then deny involvement or responsibility for unauthorized actions, undermining the accountability and trustworthiness of the application.

Information disclosure threats involve the unauthorized exposure or leakage of sensitive data within the Gen-AI application. Disclosure of confidential information, model parameters, or outputs could potentially lead to privacy breaches or intellectual property theft.

Denial of service (DoS) threats involve attacks that aim to disrupt or degrade the availability of the Gen-AI application's services. Flooding the application with excessive requests, exploiting resource exhaustion vulnerabilities, or targeting infrastructure components may render the application unavailable to legitimate users.

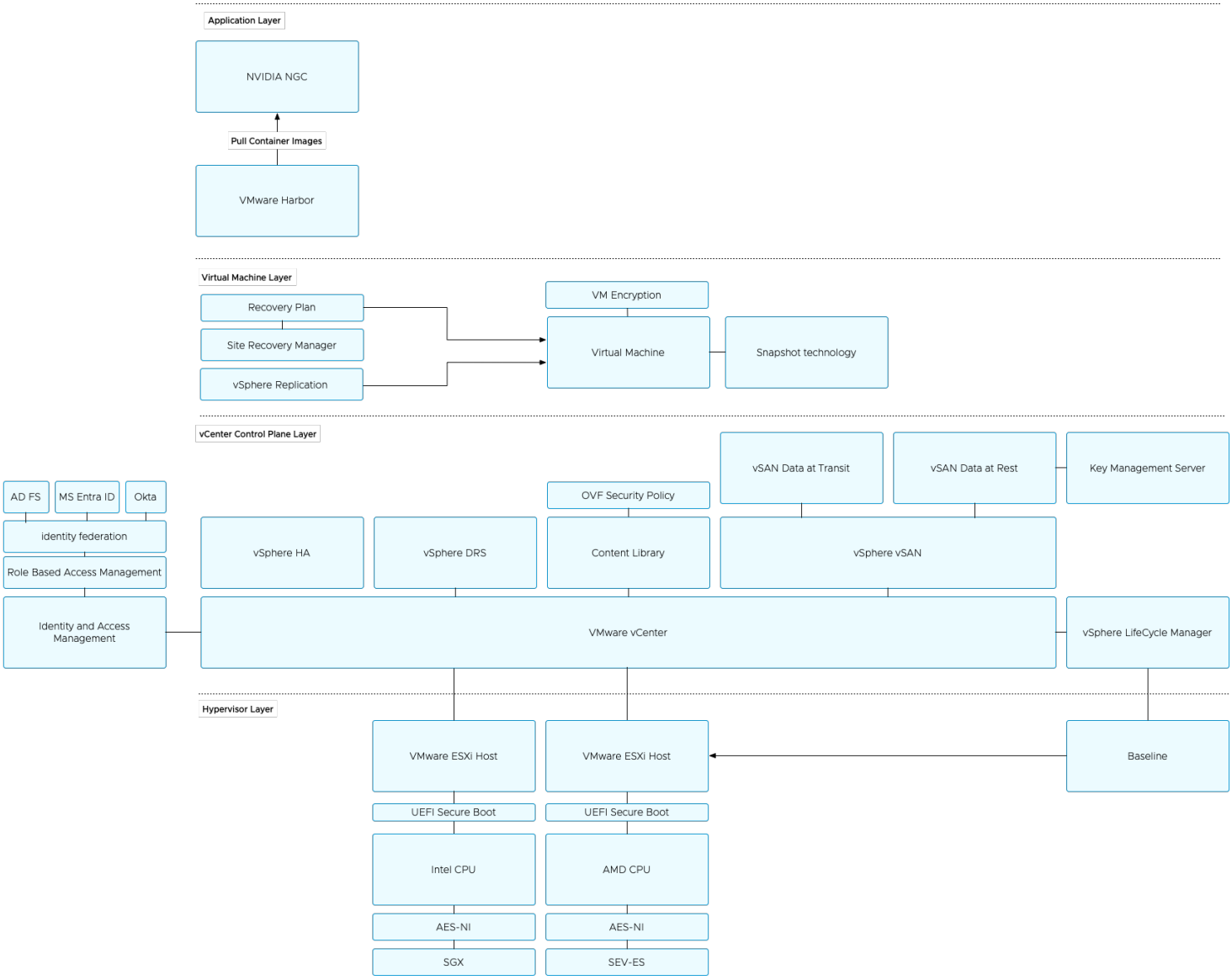
Elevation of privilege threats involve attackers gaining unauthorized access to privileged system components or resources within the Gen-AI application. Attackers escalate their privileges, bypass access controls, or perform unauthorized actions with elevated permissions, potentially leading to further security breaches or data compromises.

By considering these various threat categories outlined by the STRIDE model, organizations can develop comprehensive security strategies to mitigate risk and protect Gen-AI applications from potential security vulnerabilities and attacks.

VCF platform

Below is an overview of the components of VCF and where they reside. The components are viewed through the lens of the CIA TRIAD.

Figure 1. VCF component overview as seen through the lens of the CIA TRIAD



The CIA triad – guiding model in information security

Building on the STRIDE framework's identification of potential threats in Gen-AI applications, the CIA model offers a complementary approach to create a secure environment. STRIDE helps us understand the "what" - *the different attack vectors*. The CIA triad, on the other hand, addresses the "how" - *outlining security properties we can achieve through controls*.

This shift allows us to translate identified threats from STRIDE into actionable security measures. By applying the CIA triad security model to the infrastructure and application, we create an environment that safeguards data and functionality throughout the Gen-AI lifecycle. Securing the overall system allows the ops and data science teams to address vulnerabilities identified by STRIDE and proactively mitigate potential risks.

Traditional machine learning models are built from scratch, but foundation models—pre-trained AI powerhouses—offer a solid base for the LLM. Organizations can leverage transfer learning to fine-tune these models with their own data, creating specialized applications. While this approach streamlines development, it introduces new security considerations like data leakage, bias transfer, and single point of failure within the end-to-end pipeline.

This section will delve into the CIA triad and explore the features and functionality of the Private AI solution that helps to secure model-based AI applications, while examining security considerations throughout the entire development pipeline. Throughout this document, we will use a hypothetical Generative AI application with an LLM and Retrieval Augmented Generation (RAG) architecture as an example to illustrate the importance of security best practices. This application could represent various real-world use cases, such as a chatbot interacting with internal documentation or a fraud detection system analyzing financial transactions. By examining this hypothetical architecture, we can explore the potential security risks and demonstrate how robust security measures can mitigate them.

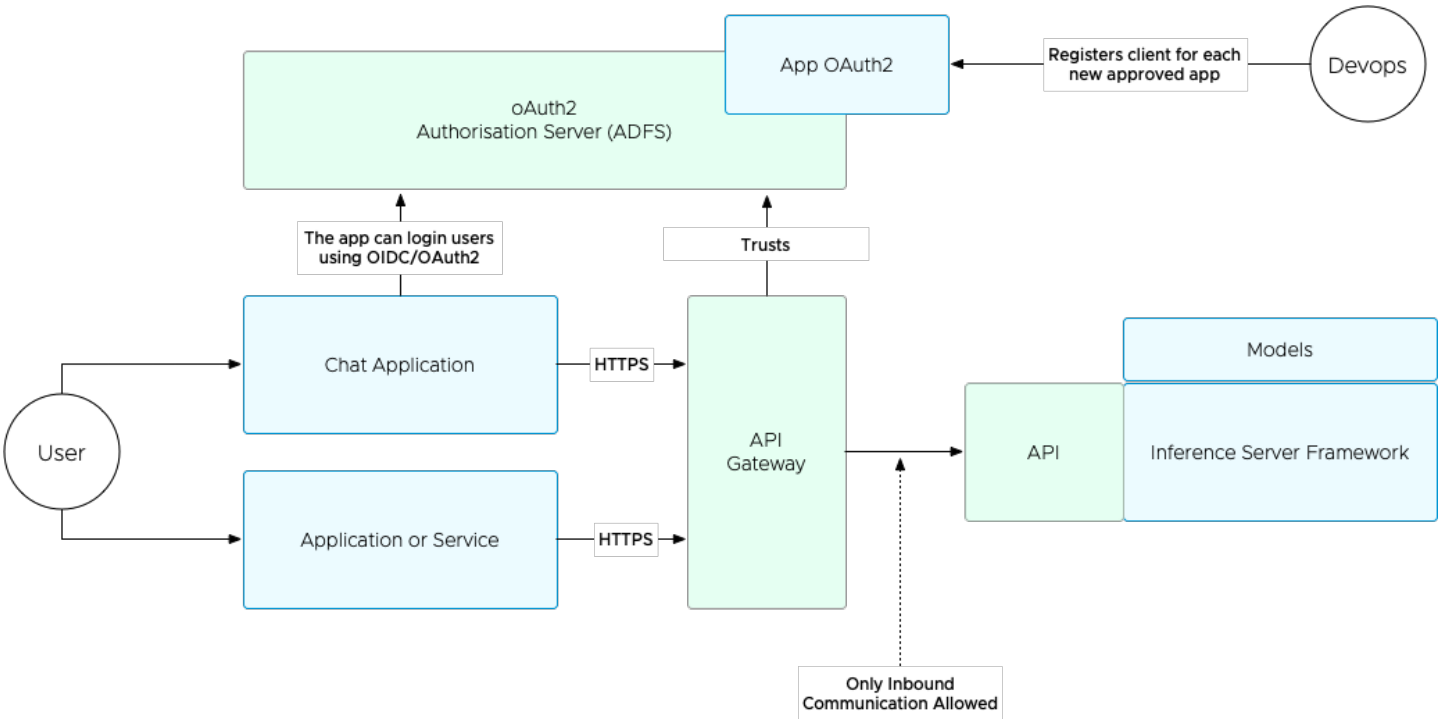
Confidentiality

Confidentiality ensures data privacy by keeping it secure and inaccessible to unauthorized users. Confidentiality involves securing sensitive data and controlling access to AI models. Information disclosure (the "I" in STRIDE) directly threatens confidentiality. Techniques like data minimization (collecting only the necessary data) and data anonymization (removing personally identifiable information) reduce the amount of sensitive data stored and processed. Additionally, implementing strong data encryption protects data even if a breach occurs.

Securing model access

Most inference server frameworks expose an API (HTTP/HTTPS, or gRPC). This API will be called by any application (chatbot, service, and application) that needs to query the models served by the server. Unfortunately, while exposing the endpoint on HTTP/HTTPS, there is no authentication or authorization. VMware highly recommends adding an API gateway in front of the model API exposed by the Inference server framework that will be in charge of authentication and authorization. API gateways typically support multiple authentication methods. We recommend using OAuth 2.0 (aka OAuth2) as the mechanism to authenticate and authorize applications that call the API gateway. DevOps or MLOps users can register and control which application is trusted to communicate with the model API.

Figure 2. Recommended architecture to secure HTTPS endpoints using an API and OAuth 2.0



User privacy controls

Data should be kept secret and private; depending on the sensitivity of data, organizations can choose to anonymize or encrypt. From a shared responsibility perspective, the data science team should focus on data minimization and data anonymization. The IT Ops team provides data encryption in transit and at rest. Let's zoom in on the various user privacy controls more closely.

Data integrity and privacy considerations ensure data accuracy and integrity. Data cleansing and validation operations by the data engineers or data scientists should prevent accidental corruption and ensure the accuracy and integrity of the data being indexed. Checksums or cryptographic hashes can be used to identify any unauthorized modifications to the data and allow the organization to detect potential tampering attempts during the indexing or retrieval processes. Data minimization and de-identification help restrict the data that is necessary for the system's functionality. By minimizing the collection and storage of unnecessary personal data, organizations adhere to responsible AI guidelines and data privacy regulations like GDPR, CPRA, and HIPAA.

Personal Identifiable Information (PII) should be redacted before embedding it in the vector database (indexing) or passing it to the LLM. Libraries such as Private AI Redact, which can be included under the LLM open-source orchestration framework LangChain, provide redaction to avoid including PII in the embeddings. Minimizing the amount of information indexed or retrieved reduces the attack surface and potential impact of a security breach.

Access control mechanisms and data encryption are essential to achieve a high level of data security. Access control mechanisms allow organizations to restrict unauthorized access to the indexed data.

Platform controls

Indexing and retrieval workflows should be encrypted for enhanced security. All data being transferred from the source to the indexing system should also be encrypted. If the data sources contain PII, VMware recommends encrypting the data at rest as well for protection against a system breach. The level of encryption required depends on the sensitivity of the data being processed. Highly sensitive data like PII or confidential organizational information warrants stronger encryption algorithms.

Data encryption

Data could be encrypted when stored on disk (data at rest) or when it is moving between systems (data in transit). Data encryption protects data if a breach occurs, avoiding or minimizing information disclosure (the "I" in STRIDE)

Private AI native support for data-at-rest encryption can occur either inside a virtual machine such as VM Encryption or can be accommodated by a storage system using vSAN data-at-rest encryption. vSAN data-in-transit encryption ensures that all data moving across ESXi hosts and file service inter-host connections in the cluster are encrypted. The data-in-transit encryption option can be enabled independently from the vSAN data-at-rest encryption, but enabling both will provide a complete, end-to-end encryption solution on a per-vSphere and vSAN cluster basis. Please note that data-at-rest encryption requires a key management server or a vSphere native key provider.

VMware achieves FIPS 140-2 validation under the Cryptographic Module Validation Program (CMVP) [6]. The CMVP is a joint NIST and Communications Security Establishment (CSE) program. FIPS 140-2 is a cryptographic module standard that governs security requirements in 11 areas relating to the design and implementation of a cryptographic module. vSphere and vSAN use the validated cryptographic module for all encryption services.

Please note that encryption adds some processing overhead. Encryption can lead to slower response times for the application, especially for tasks like indexing and retrieval where large amounts of data might be involved. Impact varies depending on the encryption algorithm and the available infrastructure. A stronger encryption algorithm offers better security but requires more processing power, leading to an increased resource requirement and slower response times. However, modern CPUs with dedicated encryption hardware can offload encryption tasks more efficiently, minimizing the impact on response times. Most of Intel Sapphire Rapids processor types have a built-in QuickAssist Technology (QAT) accelerator that offloads data encryption and decryption processes from the CPU cores.

Encrypting all data streams supporting an AI application may be overly resource intensive. The goal is to strike a balance between security and performance, underscoring the custodial advantages of a Private AI foundation deployment. Use an encryption strategy per application or per use case that can be defined and granularly invoked.

For data in transit, VMware highly recommends that data transmitted between the various Gen-AI application components and between the Gen-AI application and the client is encrypted and protected from interception by applying secure communication protocols such as Transport Layer Security (TLS).

For example, in an AI application using RAG that serves and contains sensitive data, consider minimizing the amount of information indexed or retrieved to reduce the attack surface. For retrieval workflows, only consider encryption when potentially sensitive information is served. For retrieval data at rest, most data is cached for efficiency. If this cached data contains sensitive information, encrypt it at rest to minimize risk. A clear understanding of the data involved will help you prioritize encryption for indexing and retrieval data streams based on the sensitivity and potential impact of exposure.

Access control

Organizations must incorporate well-defined authentication and authorization protocols, granular permissions based on user roles, and continuous monitoring for suspicious activity. Access control mechanisms restrict unauthorized access to the data and the model themselves. This involves user authentication and authorization systems. Identity and Access Management (IAM) is a critical aspect of Private AI Foundation security. VMware provides a range of IAM capabilities that can be used to secure access to workloads, applications, and data. Role-Based Access Control (RBAC) capabilities are natively integrated within the VCF platform for control access to resources based on user roles and responsibilities. The RBAC provides roles and responsibilities to the different personas that interact with the platform, which can help reduce the risk of unauthorized access. VCF security includes identity federation capabilities to allow administrator and operator personas to access resources using their existing corporate credentials. VCF supports Active Directory Federation Services (AD FS), Microsoft Entra ID, and Okta as external identity providers. Simplified access management improves security by reducing the need for additional usernames and passwords.

API security ensures secure communication between components of the application. Use secure communication protocols to transmit user data. Hypertext Transfer Protocol Secure (HTTPS) should be the minimum standard to prevent unauthorized access. TLS plays a crucial role in preventing man-in-the-middle attacks by encrypting the communication between the client and the Gen-AI application. TLS scrambles the data being exchanged, making it unreadable to anyone who intercepts. TLS uses digital certificates to authenticate the application endpoint the user is communicating with, ensuring the user is connecting to the legitimate endpoint and not a fake one set up by an attacker. TLS verifies the integrity of data being transmitted, detecting if any unauthorized modifications have been made to the data during transmission. With encryption, authentication, and integrity synergy, TLS makes it extremely difficult for a man-in-the-middle attacker to eavesdrop on the Gen-AI application communication, steal sensitive data, or tamper with the data.

In general, the more complex the system is, the higher the data leakage risk. When an LLM system uses an orchestration layer and relies on a vector database to store custom or proprietary data, an attacker skilled in manipulating prompts can access potential entry points. If attackers gain access to a vector database, they could potentially extract sensitive data with high accuracy. By accurately retrieving specific data points like financial information or internal documents, attackers can gain valuable insights into an organization's operation. As data scientists improve the quality of the extracted data before storing it in the vector database, the attacker has a reduced chance of retrieving incorrect or misleading information. "High accuracy" implies they can consistently extract the information they are after with a high degree of certainty. This risk is amplified if components such as agents within the system have continuous access to additional data sources.

To mitigate these risks, robust security measures are crucial. When using vector databases, access control lists (ACL) and RBAC are very important, especially when dealing with proprietary or classified data. In the scenario when data classification is at play, ensure data isolation within the vector database is based on classification level. Implement ACLs that restrict access to specific data classification for every user. Consider implementing RBAC that factors in data classification. Use the security features offered by the vector database; this might include encryption at rest and in transit, secure audit logs, and user activity monitoring.

VMware Harbor Registry, a container registry, can significantly contribute to securing containerized AI applications, because it provides built-in, role-based access control to allow granular control who can access, push, or pull container images. VMware Harbor enables the organization to define a curated catalog of container images that establishes the gold standard for an organization's AI-application development efforts. RBAC control ensures that only authorized users or applications have access to the list of curated containers.

Model security

Models, especially foundation models fine-tuned with confidential data, contain sensitive information that should not be exposed to unauthorized users. Once the model is trained, it cannot "unlearn" this data, and with the right prompt, it is accessible to any user who gains access to the model via UI or API. As the previous paragraph mentioned, it is imperative to minimize access. Organizations should also consider the attack vectors beyond the UI and API.

To ensure confidentiality of traditional ML models and foundation models, trusted execution environments (TEE) and secure enclaves isolate the model and its data from the rest of the system, making it harder for unauthorized access. One of the main challenges in serving confidential machine learning models, whether it is a traditional model or an LLM, is the need to keep data encrypted, including during computing. Unfortunately, this phase is vulnerable to memory dumps. VMware makes the most secure hypervisor as demonstrated by the Common Criteria certification. However, application secrets are stored in system memory, CPU cache, and possible GPU memory. As a result, the hypervisor, the guest OS, and the application can access this information.

vSphere offers multiple functions to create secure enclaves on modern CPUs. Organizations can leverage technologies like Intel Software Guard Extensions (SGX) or AMD Secure Encrypted Virtualization-Encrypted State (SEV-ES) to create secure enclaves within the virtual machines. These enclaves isolate the fine-tuning process and model inference, protecting the confidentiality of the data and the model parameters. At VMware, we are committed to the future of data security. We're actively exploring innovative solutions—like confidential computing—that provide organizations with a more secure way to process sensitive data.

Integrity

Gen-AI applications, unlike most traditional software, are constantly evolving. These applications are trained on massive datasets and can adapt their behavior over time. This dynamic nature makes it challenging to maintain a clear understanding of the components and potential vulnerabilities within the system. Software Bill of Materials (SBOMs), play a crucial role in securing AI applications by enhancing their transparency and traceability. An SBOM acts as a detailed inventory of all the software components used to build the Gen-AI application, including open-source libraries and frameworks, pretrained models, the datasets used to fine-tune models, and the data sources used by the model and the custom code elements. For example, if one of the data sources is removed from a RAG architecture, it changes the output of the Gen-AI application. By having a comprehensive SBOM of all

the data, model, containers, VMs, and code base associated with the RAG app, organizations can gain a clear understanding of a particular AI application's composition. This transparency is vital for security purposes and threat management where SBOMs can be linked to vulnerability databases. When a security flaw is identified in a component listed within the SBOM, organizations can be promptly alerted and take necessary actions to patch the vulnerability in their Gen-AI application. Most Gen-AI applications often rely on pre-trained models and third-party code. SBOMs can help identify potential security risks introduced through these external components, allowing organizations to assess the trustworthiness of their software components.

Data integrity ensures that information generated by neural network models is trustworthy and untampered with, guaranteeing its authenticity, accuracy, and reliability. VCF provides secure boot functionality on the ESXi host to ensure only authorized operating systems are loaded. Use vMotion for live migrations of VMs with minimal downtime to maintain data integrity during infrastructure maintenance or looming hardware failures. Implement VM snapshot technology with version control systems to create rollback points in case of model corruption or errors and track changes to the model code. This will ensure traceability and integrity when fine-tuning datasets.

VCF provides multiple mechanisms for tamper detection. vSphere content libraries allows organizations to maintain a collection of approved virtual machine images for deep learning VM and Tanzu Kubernetes Grid Service distributions. VCF offers template (OVF) security policies that enforces the strict validation of OVF items during template deployment, update, import, or synchronization. Organizations can also add the OVF signing certificate from a trusted certificate authority (CA) for added security.

VMware Harbor Registry provides the means to sign and verify container images. This functionality ensures images are scanned and free from vulnerabilities, avoiding the scenario where an attacker could inject malicious code to damage the system. Add third-party software like Cosign to use signatures to verify which images are safe to use and which are compromised.

VCF includes many features to implement continuous monitoring of the environment. VMware Aria Operations enables organizations to thoroughly monitor the performance, health, and security of ESXi hosts, VMs, and the underlying infrastructure to detect any anomalies or security threats.

Infrastructure components must be regularly audited for security reasons. vSphere Lifecycle Manager (vLCM) allows organizations to ensure that the ESXi hosts, where the Gen-AI application is deployed, are provisioned and configured securely. vLCM maintains a consistent and secure baseline configuration across all ESXi hosts in the VCF environment, reducing the risk of vulnerabilities or misconfigurations. vLCM provides the host-based image functionality to customize specific configurations, patches, and GPU drivers required for the Gen-AI application. The image health and compliance checks within vLCM ensure that the ESXi hosts adhere to security and configuration standards. vLCM monitoring capabilities flag deviations from the predefined baselines and offer remediations of any non-compliant ESXi hosts. Cluster-aware updating orchestrates the update process of ESXi clusters that host the Gen-AI application in a coordinated fashion, minimizing the downtime and maintaining workload availability.

Availability

Platforms, networks, and applications must function consistently to provide uninterrupted access to information. VCF offers a powerful suite of services to guarantee the continued availability of AI applications even during outages.

The core of VMware's offerings is vSphere High Availability (HA). This feature groups ESXi hosts into clusters, constantly monitoring their health and the VMs they run. If a host fails, vSphere HA seamlessly restarts affected VMs on a healthy host within the cluster, minimizing downtime for AI applications. With the help of Dynamic DirectPath I/O, vSphere provides a flexible way to assign hardware accelerators to workloads, identifying the hardware accelerator by attributes rather than by its hardware addresses. This abstraction level allows both DRS and HA to select the appropriate ESXi host with similar hardware for workload initial placement and failure recovery.

VCF also addresses disasters. Site Recovery Manager and vSphere Replication work together to provide site protection for AI applications within VCF. These tools help IT operations teams identify critical components and data associated with AI workloads. They then replicate this vital information to a designated recovery site, ensuring a swift restoration of applications after a major outage.

Resource optimization is another key benefit of the VMware platform. The Distributed Resource Scheduler (DRS) works in tandem with vSphere HA to intelligently distribute VMs across available ESXi hosts in a cluster. Distribution optimizes resource utilization, preventing any single host from becoming overloaded and hindering AI application performance. DRS also mitigates the impact of other demanding workloads on your AI applications. If a host failure occurs, DRS selects the most suitable healthy host to minimize disruption to your AI workloads.

Data availability is ensured through shared storage. VCF supports a variety of shared storage solutions that provide block-level, file-level, or object-based storage access to VMs and containers. These systems can be configured to mirror and replicate data, maintaining availability even during storage or host failures.

Live migration is enabled by vMotion, a vSphere feature that enables live migration of running applications between ESXi hosts within a cluster. vMotion decouples infrastructure maintenance from application uptime, enabling near-zero downtime for your AI applications.

Finally, VMware addresses network availability. Within vSphere, ESXi hosts can be configured with multiple redundant network paths to ensure network availability even if one path fails. Additionally, VMware NSX, a network virtualization platform, pairs with vSphere HA to improve failover times for AI applications. NSX automatically replicates its configuration across multiple managers, minimizing disruption to network traffic.

VCF's comprehensive features provide a robust solution for maintaining the availability and performance of your AI applications, ensuring continuous and reliable access to valuable information.

Air-gapped environment

The goal of a disconnected or air-gapped environment is to isolate sensitive and critical systems from other networks. The purpose of an air-gap architecture is to create a barrier that prevents unauthorized access, data breaches, or attacks from spreading to or compromising the isolated systems.

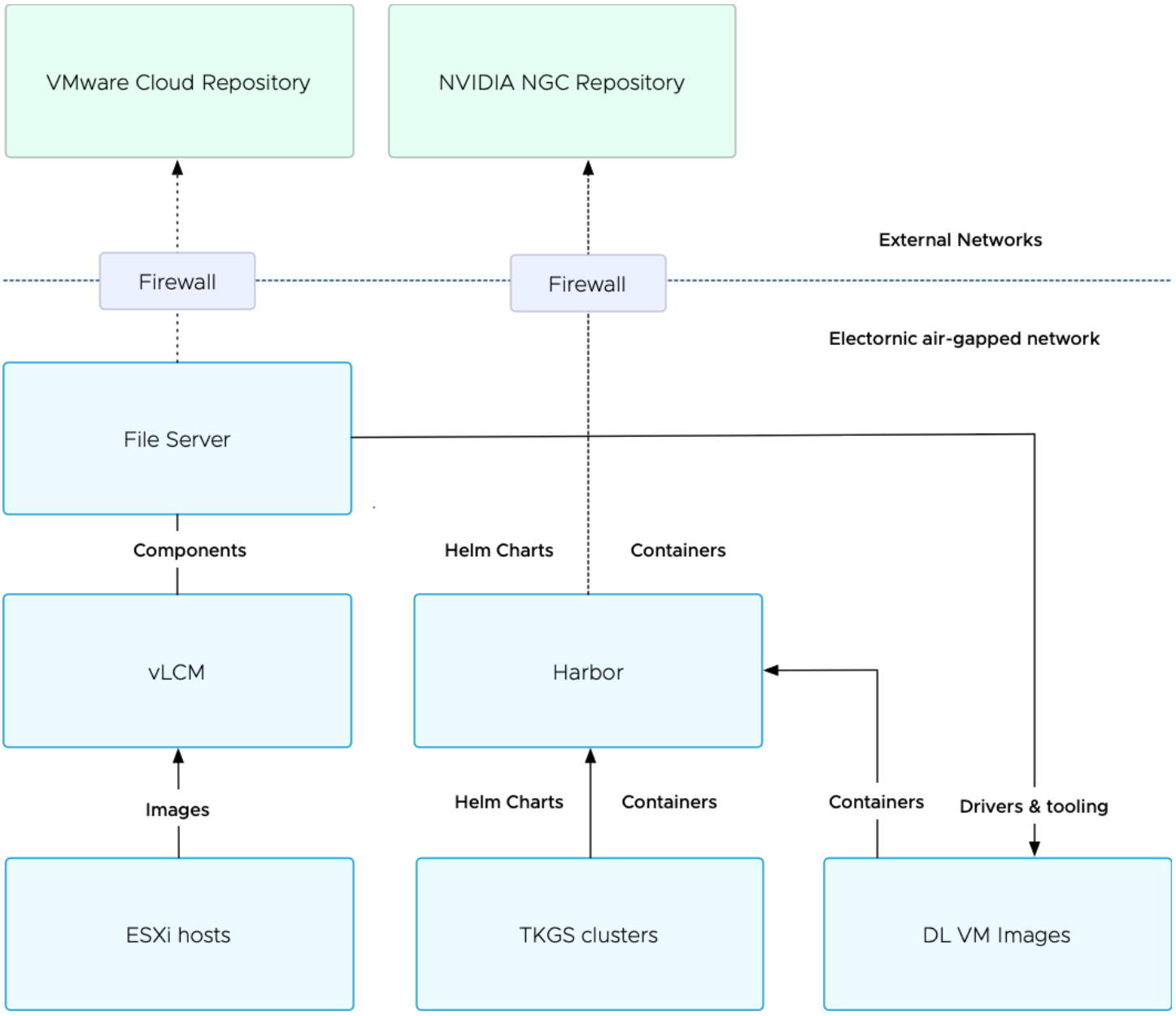
Air-gapped environments, while offering high security by physically isolating a network from external connections, come with their own set of challenges, such as limited entry and exit of data transfers, significant complex data sanitation procedures, and challenging lifecycle management. Updating software and patches in an air-gapped environment becomes a manual process. Patches need to be downloaded from a trusted source outside the air-gapped environment, scanned for malware, and then transferred securely into the environment physically. This is time consuming, resource-intensive, and can introduce other attack vectors due to exploitations in removable storage mediums, clearly demonstrated by Stuxnet [7].

To deal with or circumvent this complexity, most organizations use an "electronic air-gapped" network reinforced by a zero-trust security model. The network is deliberately isolated from external networks, but it allows external connections through a firewall that filters traffic in both directions (inbound and outbound) to limit traffic only to what is required. Electronic air-gapped networks are disconnected from external networks most of the time. Connections are opened by manually changing the firewall policies at particular times or dates. A very sophisticated security method is time-based connectivity, which is when the firewall filters are activated based on predetermined times and dates. This is to automate software update downloads. Since most software stacks are composed of software from multiple software vendors, it's very difficult to minimize the external connectivity due to the different software release patterns.

Customers should perform regular maintenance tasks, such as software updates, patch management, and downloads of new foundation models to ensure the continued reliability and security of the electronic air-gapped environment. An SBOM can play a critical role when the electronic air-gapped environment is opened up for controlled updates or data exchange. An SBOM streamlines the process of identifying necessary updates and potential security risks associated with introducing external components. Although SBOMs lose their vulnerability detection edge in electronic air-gapped environments (because there is no external connection to access these vulnerabilities' databases), the SBOM can still promote better internal hygiene and serve as a foundation for future secure integration.

To minimize external connectivity moments, we recommend using local software repositories and offline licensing functionality. VMware Harbor Registry is our preferred method for the private container registry providing images and Helm charts from internal and external sources like NVIDIA GPU Cloud. The NVIDIA Delegated License Service (DLS) instance can be hosted on-premises. The DLS instance stores NVIDIA AI Enterprise (NVAIE) licenses offline. The licenses should be downloaded from the NVIDIA licensing portal. VMware deep learning VM templates allow customers to quickly start AI projects and experiments. The DL VM provides the ability to pull NVIDIA NGC containers into the VM. In an electronic air-gapped environment, a file share must be available containing the GPU driver in order for the template installation to be completed. A local container registry (VMware Harbor Registry) should be available to host the NGC containers.

Figure 3. Architecture of an air-gapped environment for VCF



Securing the Gen-AI application

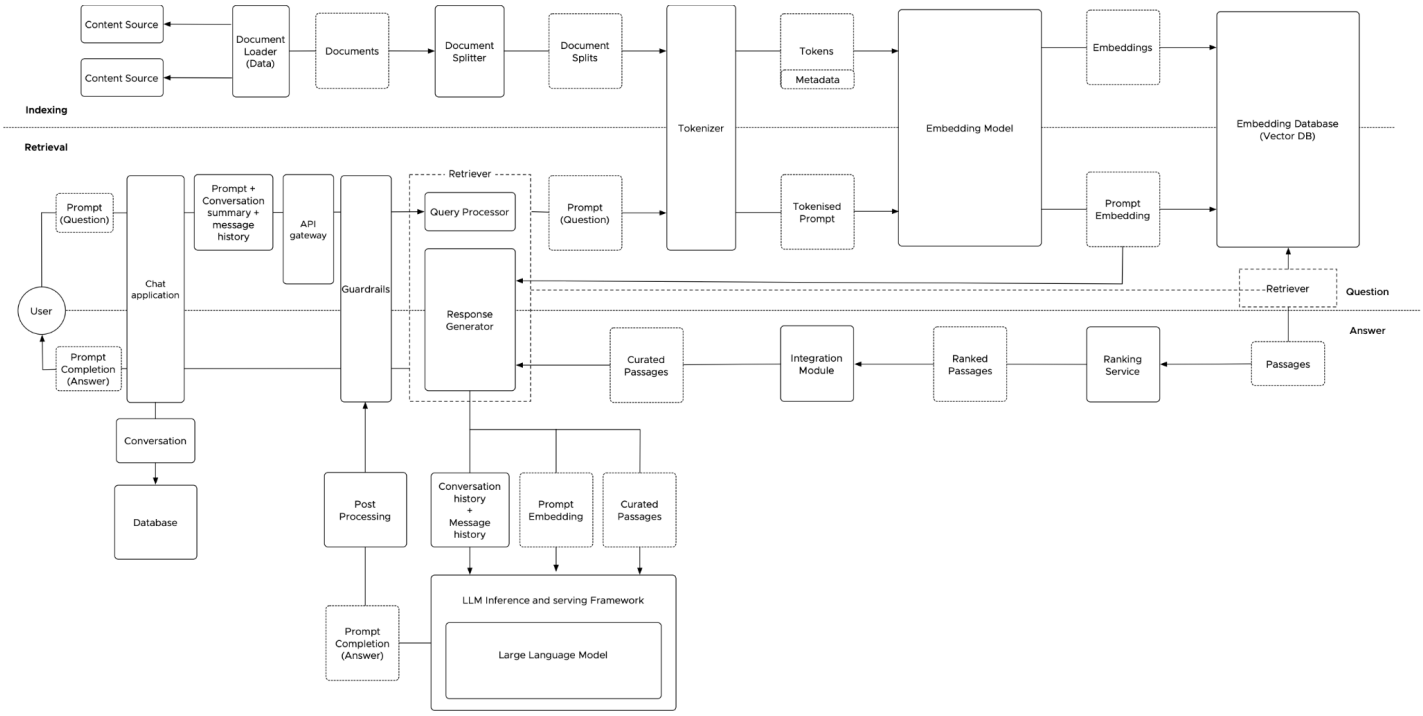
Securing LLMs and RAG architecture is incredibly important in today's data-driven and interconnected landscape. As these Gen-AI applications become increasingly powerful and common throughout the datacenter, ensuring their safety, integrity, and availability is of utmost importance for the entire enterprise.

Retrieval Augmented Generation (RAG) is a technique for augmenting LLMs' knowledge with additional data. In a standard Gen-AI application using an LLM as its sole knowledge source, the model generates responses based only on the input from the user query and the knowledge it has been trained on. The application does not actively retrieve additional information beyond what is encoded in its parameters during fine-tuning or training. In contrast, an RAG architecture integrates both retrieval-based and generative components. It includes a retriever component that obtains relevant information from a large collection of text (typically referred to as a corpus). This corpus is stored in an embedding database, commonly referred to as a vector database. The retrieved information is then used by the generative component to produce responses. Let's take a closer look at the indexing and retrieval process and explore the roles of the various components in the key stages of retrieval augmented generation.

Full end-to-end workflow of a RAG architecture

Figure 4 diagrams the components, their input, and their outputs (elements with dotted lines) of a RAG architecture. Most data scientists design such a system with the help of frameworks such as LangChain or LlamaIndex. These frameworks provide functionality, as depicted below, but sometimes without having distinct components for each individual task. This diagram shows the conceptual tasks and components used in the indexing and retrieval processes. Its primary goal is to highlight the different phases data go through and the shared components used by both processes.

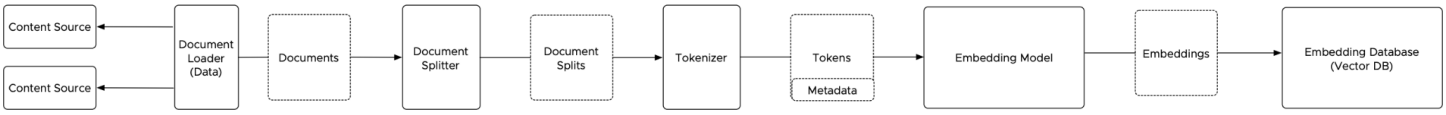
Figure 4. End-to-end workflow of a RAG architecture



Building the foundation for retrieval: the indexing process in RAG architectures

The indexing stage in a RAG architecture lays the groundwork for efficient information retrieval. It involves transforming a vast collection of data sources, regardless of structure (unstructured documents like PDFs, semi-structured data like JSON, or structured data from databases), into a format that LLMs can readily use. This process can be broken down into a Load-Transform-Embed-Store workflow.

Figure 5. Example of the indexing process in a RAG architecture



Indexing process

Loading diverse data sources

The indexing process begins with data loaders, which act as information gatherers. They retrieve data from various sources, including unstructured documents (for example, PDF and DOC files), semi-structured data (for example, XML, JSON, and CSV files), and even structured data residing in SQL databases. These loaders then convert the retrieved data into a standardized document format for further processing.

Transforming data for efficient processing

Document splitters take the stage next. They play a crucial role in organizing and preparing the data for efficient processing by the embedding model. They segment the documents into logical units—sentences or paragraphs—based on predefined rules. This segmentation ensures that information remains semantically intact while preparing it for further processing. Imagine a large research paper being fed into the system. The document splitter receives the PDF from the loader and meticulously splits it into individual paragraphs for further processing.

Tokenization – the building blocks of meaning

Following segmentation, the tokenizer steps in. It takes each logical unit (e.g., paragraph) from the document splitter and breaks it into its fundamental building blocks: tokens. These tokens can be individual words, sub-words, or even characters, depending on the chosen embedding model and the desired level of granularity. Accurate tokenization is critical for tasks that rely on understanding the meaning of the text, as it forms the basis for how the LLM interprets the information. Since the tokenizer essentially defines the vocabulary understood by the entire RAG architecture, a single shared tokenizer process across all components dealing with text processing and encoding is required. Choosing an embedding model dictates the tokenizer you must use within this architecture. These two components are deeply connected.

Embedding – capturing semantic meaning

Once tokenization is complete, the embedding model takes center stage. The model's role is to convert each token into a numerical vector representation, capturing its semantic meaning within the context of the surrounding text. Pre-trained embedding models, either word embeddings or contextual embeddings, achieve this by mapping the tokens into these vector representations.

Finally, an indexing component takes over. It packages the generated embedding vectors along with any associated metadata (for example, document source information) and sends them to a specialized embedding database—the vector database (vector DB)—for efficient storage. This database becomes the foundation for the retrieval stage, where the RAG architecture searches for relevant information based on user queries.

The stored foundation

The vector database plays a crucial role in efficient retrieval. It stores the embedding vectors in a three-dimensional space, allowing for fast and effective search operations based on vector similarity. The embedding model paves the way for the retrieval process, where the RAG architecture efficiently locates relevant information from the indexed data based on user queries, ultimately enabling the LLM to generate informative and relevant responses.

Data preparation and security

Within a RAG architecture, data quality and integrity are crucial. RAG systems function best when the information is reliable. Inaccurate or misleading data can disrupt the retrieval process, potentially feeding irrelevant information into the augmentation and generation stages, leading to outputs that are nonsensical or irrelevant. Therefore, maintaining clean and trustworthy data is essential for a RAG system to function effectively. Data scientists have a number of tools and techniques at their disposal to ensure and improve data quality and integrity throughout the entire data lifecycle.

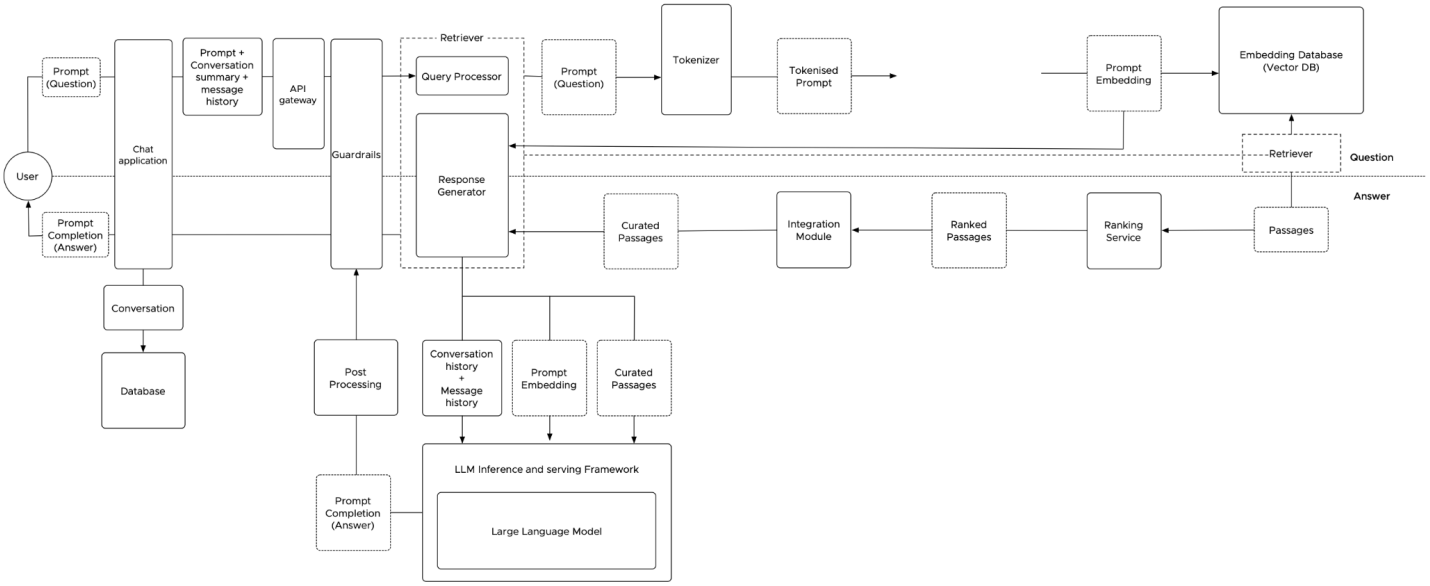
During the data preparation stage, the raw data sources (for example, Confluence pages) are systematically gathered. Maintaining the original accessibility levels of these pages by mirroring the initial access control roles is crucial at this point. In scenarios involving document segmentation, associate these role-based permissions with each individual chunked embedding stored in the vector database. This step upholds the integrity and confidentiality of the information.

When processing a query, the system first verifies the role of the user. The system strictly grants access in accordance with these roles, enabling the user to access only those tables or specific rows within the vector database for which they have clearance. The documents that were retrieved provide contextual data for the LLM. Additionally, only authorized users can access the LLM itself. If the conversation history is stored, access must be restricted to either the user or the group of users conducting the conversation.

Retrieval – efficiently finding relevant information

The retrieval stage in a RAG architecture is where the magic happens. To fuel the LLM generation capabilities, the system efficiently locates relevant information from the indexed data. The system processes the user's query, often referred to as a prompt in natural language processing, using the same "language" for creating and storing the embeddings during indexing.

Figure 6. Retrieval stage example in a RAG architecture



Retrieval process

Understanding user queries

The process begins with the user submitting a query, often phrased as a natural language prompt (question, instruction, and so on). Before the user can generate a query, the organization must ensure that only authorized users can use the RAG system. When the embedding model is filled with confidential data, it becomes your one-stop shop for company secrets. For private AI purposes, the API gateway acts as a central entry point for "external" requests and queries, handling tasks such as authentication, rate limiting, and request logging, and ensuring that interactions with the RAG system are secure and well documented.

Authentication ensures that only authorized users and applications can access the Private AI system. The components within the entire RAG architecture should use the same identity provider to create a single security domain. This allows the API gateway to verify the user identities and ensure they have the necessary permissions. The API gateway routes requests to specific components or microservices within the RAG architecture based on predefined criteria. Data flow management controls the flow of sensitive data and ensures it only reaches authorized components. The API gateway's logging and monitoring functionality allow it to log all incoming and outgoing traffic, providing valuable information for security audits and monitoring of potential threats. Many API gateways implement caching mechanisms to improve response times and reduce the load on internal components, especially for frequently accessed data.

Inherently, LLMs are stateless; therefore, the LLM itself does not track the "conversation" between itself and the user. Conversation memory allows the chat application to store past interactions and user preferences, providing context for the LLM when generating responses. Using conversation memory leads to more relevant, coherent, and engaging interactions. Most chat applications store the conversation memory in a traditional database, depending on their implementation. Conversation memory, along with the prompt to reference past interactions, are sent, which allows the vector database to enhance its similarity search and the LLM to tailor the response to the ongoing conversation.

The prompt must be translated into the same format used to create and store the embeddings during indexing. To achieve this, the system leverages the same tokenizer and embedding model employed in the indexing stage. The tokenizer breaks the prompt into tokens (words or subwords) and then converts it into a vector representation using the pre-trained model. This vector representation captures the semantic meaning of the prompt within the context of the larger language model.

Before processing the prompt further, it's essential to apply guardrails to ensure it meets certain safety, ethical, and quality standards. Guardrails play a crucial role in preventing misuse and ensuring that the generated responses align with the organization's ethical guidelines and expectations. Guidelines like the OWASP Top 10 for LLM Applications and Meta's Llama 2 Responsible Use Guide recommend that generative AI applications implement guardrails throughout the system. Guardrails examine both the information fed into the model by human prompts (inbound safeguard) and the output the LLM generates (outbound safeguard). These guardrails help prevent the model from violating organizational policies, protect against manipulation by malicious users, and avoid generating harmful content.

Guardrail implementation approaches can be categorized into two main types: library-based safeguards and fine-tuned LLMs. Library-based safeguards are open-source toolkits that provide pre-built functionality for implementing guardrails and are easily integrated into existing LLM systems, reducing development time and necessary expertise. Most library-based safeguard solutions like Safeguard AI or NVIDIA NeMo provide common guardrail functionalities like filtering outputs for specific criteria (hate speech, bias) or guiding the LLM to stay on topic.

Guardrail mechanisms for fine-tuned LLMs involve creating a separate LLM that is specifically designed to analyze and potentially adjust the outputs of another LLM. This two-tiered model is more complicated than library-based safeguard solutions, and the fine-tuned LLM itself can inherit bias if not carefully trained on unbiased data. The two-tiered model can provide more sophisticated control over the outputs because it can potentially analyze the context in which the main LLM is generating text, leading to more nuanced guardrails. An example of this is the Llama Guard LLM, which is a 7-billion-parameter Llama 2–based input-output safeguard.

Matching queries with encoded information

With the query transformed into a vector, the retrieval process can efficiently search through the collections of embeddings stored in the vector database. This search hinges on the principle of vector similarity—the system seeks embeddings within the database that closely resemble the prompt's vector representation. These retrieved embeddings, referred to as passages, typically represent relevant extracts/sections from the indexed data.

Prioritizing relevant passages

Not all retrieved passages hold equal weight. A ranking service prioritizes the most relevant ones. This service applies a ranking algorithm that considers factors like the the degree of similarity between the passage's embedding and the prompt's vector to assign a score to each retrieved passage. This scoring identifies the passages most likely to address the user's query.

Preparing information for the LLM

The integration module is the bridge between the retrieved information and the LLM. The module receives the ranked passages and performs crucial formatting tasks. Depending on the specific task and the system design, the integration module might employ summarization techniques to condense lengthy passages or use answer extraction methods to pinpoint the most relevant information within a passage. In some scenarios, the module might select a single top-ranked passage for processing (single-passage processing), while others might leverage multiple high-ranking passages (multi-passage processing). The integration module then prepares these passages, potentially concatenating or processing them individually to align with the input format expected by the LLM.

Feeding the LLM

Finally, the integration module presents the prepared passages alongside the embedded prompt to the LLM. Now empowered to process the information, the LLM draws upon its knowledge and understanding of language to generate a comprehensive and informative response that aligns with the user's query.

Going from generation to user experience – the final steps in RAG

The journey of a RAG response continues after the LLM generates its initial output. Several vital steps ensure the user receives a refined, informative, and well-presented response. This stage encompasses post-processing, formatting, user interface integration, and, ultimately, user presentation.

Polishing the response – post-processing

The raw output from the LLM might undergo some post-processing steps to enhance its quality. Before the user can generate a query, the organization must ensure that only authorized users can use the RAG system. When the embedding model is filled with confidential data, it becomes your one-stop shop for company secrets. For private AI purposes, the API gateway acts as a central entry point for "external" requests and queries and handles tasks such as authentication, rate limiting, and request logging, ensuring that interactions with the RAG system are secure and well-documented. This could involve tasks like:

- **Text normalization**: Ensuring consistency in formatting, such as converting all numbers to a standard format or handling special characters.
- **Spell checking**: Identifying and correcting any potential typos or spelling errors.
- **Grammar correction**: Refining the grammatical structure of the generated text for clarity and coherence.
- **Redundancy removal**: Eliminating unnecessary repetition or irrelevant information that may clutter the response.

These post-processing steps ensure the generated response is informative, grammatically sound, and easy for the user to understand. Additionally, guardrails can evaluate and filter the generated outputs to ensure they meet predefined criteria. This may involve automated checks for compliance with safety, ethical, or quality standards, as well as human review processes to verify the suitability of the generated content.

Tailoring the response for presentation

The generated response might need formatting adjustments before presentation, depending on the application and user interface requirements. This formatting could involve structuring and adding visual elements. When structuring the response, the content is organized into well-defined paragraphs for improved readability. Post-processing adds visual elements such as bullet points, headers, or even multimedia content (if applicable) and enhances clarity and user engagement.

Seamless integration – user interface and presentation

Once processed and formatted, the response is seamlessly integrated into the application or platform's user interface. This user interface could be a web page, mobile app, chat interface, or any other medium through which users interact with the system. This integration ensures a smooth flow of information from the RAG architecture to the user experience layer.

User presentation and interaction

Finally, the polished and formatted response reaches the user through the chosen interface. The user can then review the information, provide necessary feedback, or take further actions based on the response. Depending on the application, users can interact further with the system by asking follow-up questions or initiating new tasks.

Maintaining a positive user experience

Throughout this final stage, user experience remains the critical focus. The generated response should meet the user's accuracy, relevance, and readability expectations. Additionally, error-handling mechanisms should address any potential issues that arise during response generation or presentation. User feedback loops can also be implemented to continuously improve the performance of the RAG model and deliver consistently valuable experiences.

By effectively managing these final steps, RAG architectures can not only generate high-quality responses but also ensure responses are presented in a way that maximizes user satisfaction and understanding.

Private AI network security

While some threat actors are looking for a quick payday, cyber-espionage groups may seek entry into an enterprise infrastructure and live off the land (LOTL) for extended periods of time, sometimes going longer than 5 years. These groups are persistent in their desire to find high-value assets and then slowly apply tactics, techniques, and procedures (TTPs) to create disruption and introduce uncertainty. Attackers are already using AI tools to facilitate their attacks. As organizations begin to develop and deploy Private AI architectures, it is only a matter of time before attackers find ways to compromise those architectures.

On one hand, like any other workload, Private AI workloads are just virtual machines and containers that support the business goals of the enterprise. On the other hand, Private AI workloads are high-value targets for malicious threat actors that provide the enterprise with business-critical data. Therefore, it is critical to give special attention to the security surrounding these workloads.

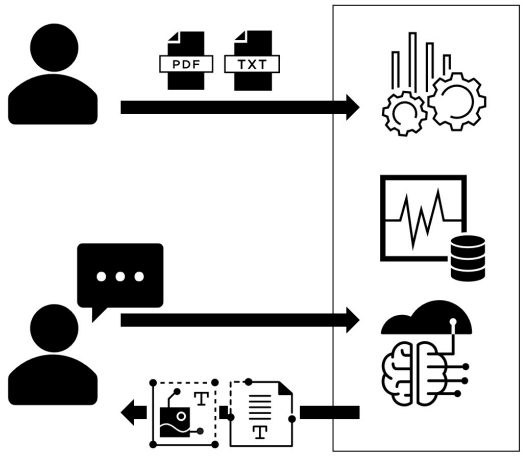
Securing Private AI workloads is done at three levels:

1. Securing ingress traffic to the AI workloads
2. Securing intra-AI component communications
3. Securing egress traffic from the AI workloads

Securing Private AI ingress traffic

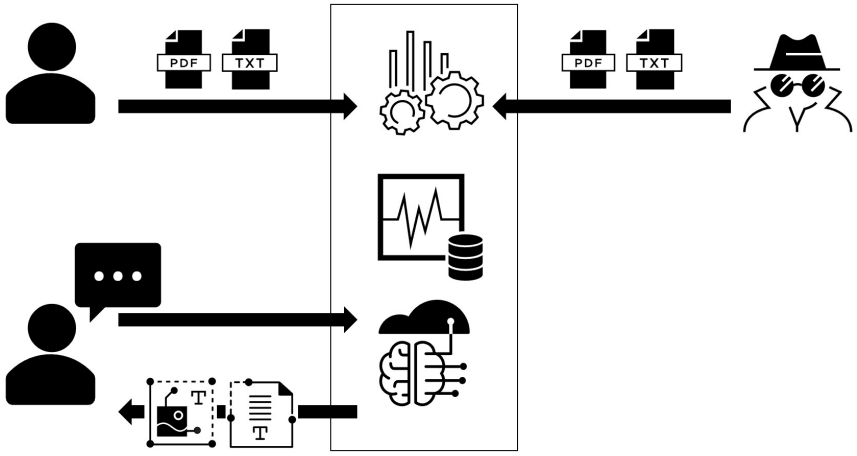
When it comes to AI models, the quality of the data provided to the AI model is directly related to the quality of data that will come out of the model. The better the data quality that goes in, the better the data quality that comes out. Likewise, when low quality or malicious data goes in, the data coming back is misleading or simply wrong. The users interacting with a Private AI platform assume that the results returned were generated by an LLM trained with valid data provided by an authorized individual, as shown in figure 7.

Figure 7. Users interacting with a Private AI platform inherently trust the results of their interaction, presuming the model was trained with valid data submitted by an authorized individual.



Private AI architecture that is not secured against malicious data upload compromises the results returned to the users. Figure 8 exemplifies a threat actor that has access to the Private AI training components. With open access, the threat actors can submit unauthorized documents to train the LLM. Users interacting with the Private AI system are now getting compromised or bad results while still assuming that the source of truth was an authorized individual or data source.

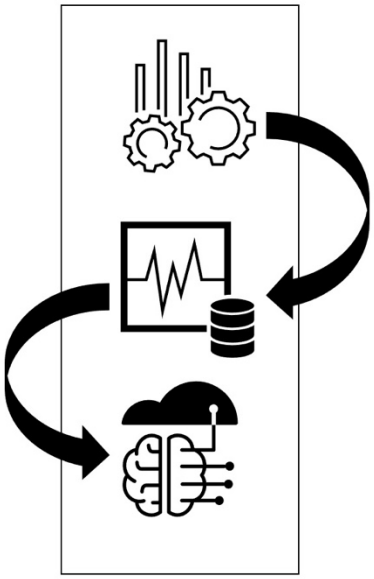
Figure 8. A malicious threat actor gains access to Private AI training components. They compromise the quality of the results, returned to users who assume the legitimacy of the source documents.



Securing Private AI ingress traffic

AI architectures are made up of many components. Each of those components, when unprotected, represents a potential vulnerability to the overall security posture of the application's deployment.

Figure 9. Lateral movement between AI components should be controlled to limit communications to only the necessary sources, destinations, ports, and protocols.

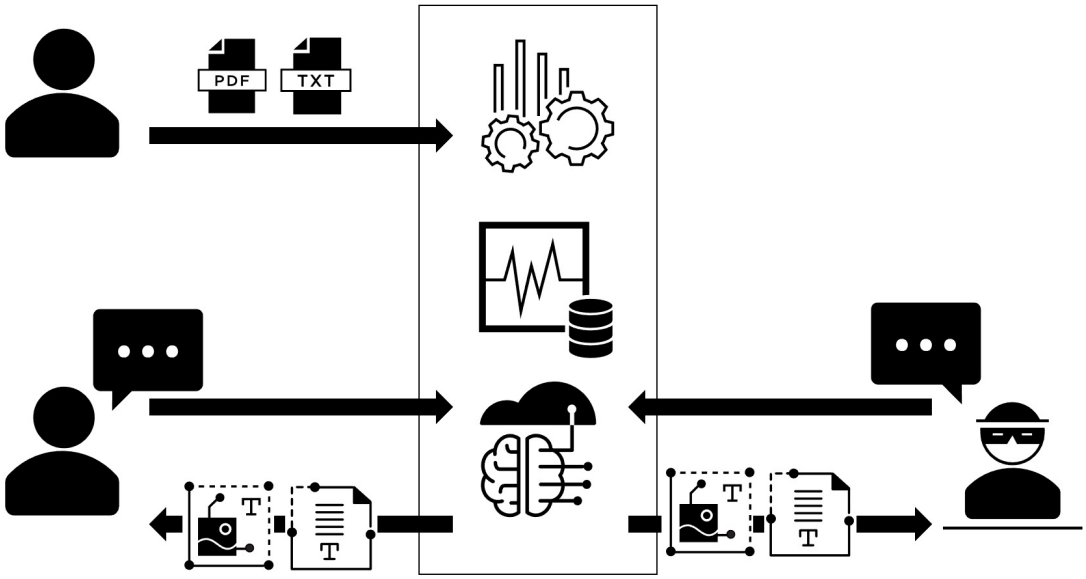


Establishing communication controls between the AI architecture components requires meticulous planning and detailed understanding of how each component communicates with every other component, if at all. In a later section, this paper will cover the process of discovery and protection for intra-AI component communication.

Securing egress traffic from Private AI workloads

Since a Private AI environment eases users' access to company or customer information to support their jobs, access to the information should be controlled to prevent unauthorized individuals from accessing proprietary data. Allowing unauthorized access to the AI data using common language, shown in figure 10, makes it easy for threat actors to find information that provides either direct financial benefit or offers sensitive information that is beneficial to long-game corporate espionage attacks.

Figure 10. Having unauthorized access to the AI application, malicious users can easily discover sensitive/close-hold/proprietary information about an organization.

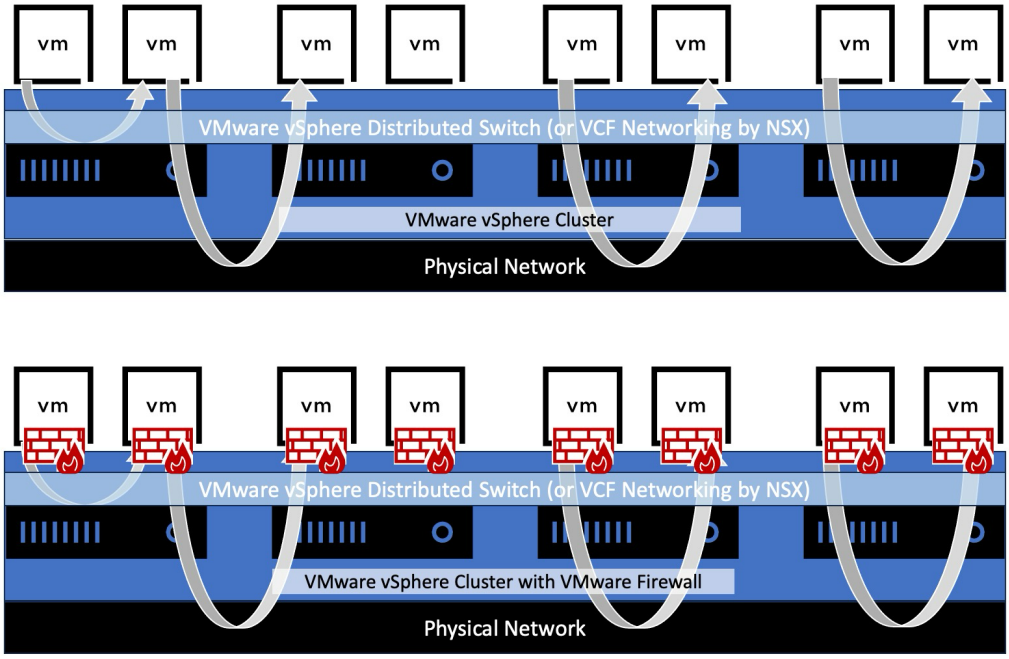


VMware vDefend

VMware vDefend includes VMware vDefend Distributed Firewall, VMware vDefend Gateway Firewall and VMware vDefend Intelligence. VMware vDefend is purpose built to create unparalleled security enforcement for the VMware private cloud.

The VMware vDefend Distributed Firewall acts like a traditional layer 7 firewall to protect traffic flows, with the added advantage that it is built into the vSphere hypervisor. Distributed switches create VLAN-backed port groups or software-defined networking logical networks that enable the hypervisor to simply pass the IP traffic in your environment. The enhanced security capability of the VMware Firewall to the vSphere hypervisor adds inspection of that same traffic against a set of rules to determine if it should be permitted or denied. VMware vSphere enforces VMware Firewall policies at the virtual NIC (vNIC) level, effectively giving each vNIC a stateful layer 7 capable firewall that inspects traffic on both traffic ingress and egress. The VMware Firewall architecture isolates the protected (the workload) from what is doing the protection (hypervisor). This isolation/separation/distinction creates a security boundary that outperforms agent-based or traditional firewall-based security options. Impact on the underlying compute infrastructure is negligible due to good security policy design and will not jeopardize application performance. Figure 11 details the VMware Firewall architecture.

Figure 11. VMware vDefend Distributed Firewall extends vSphere capability beyond just moving packets to enabling the hypervisor to inspect traffic against a set of defined security policies enforced at the vNIC level for every workload.



The hypervisor-embedded VMware Firewall policies control traffic across clusters, within a cluster, across hosts, and even within a given host. Configured policies prevent communications between workloads of the same IP subnet running on the same host. Security enforcement from within the hypervisor gives unprecedented power to control lateral movement within the private cloud.

VMware vDefend Gateway Firewall is an extension of the VMware vDefend Distributed Firewall that can be deployed as a virtual or physical appliance used to protect physical workloads or create zone-based security solutions. Together with the VMware vDefend Distributed Firewall, the private cloud can be secured through a single management plane.

In addition to protecting virtual machine workloads, the VMware Firewall solutions can provide security enforcement to container workloads. To improve the security posture of containers, VMware vDefend Distributed and Gateway Firewall permits the creation of generic groups with Kubernetes member types in dynamic membership criteria to match traffic entering into or leaving from Antrea Kubernetes clusters. These generic groups are then used in firewall rules to secure traffic between VMs in the NSX environment and pods in Antrea Kubernetes clusters.

The deep integration between VMware vSphere and VMware vDefend creates a rich contextual awareness of the virtualized environment allowing VMware vDefend security policies to be defined by elements beyond just an IP address. Policies can be built using VM name, NSX Segment, Segment Port, Distributed Port Group, IP Set, or NSX Tag.

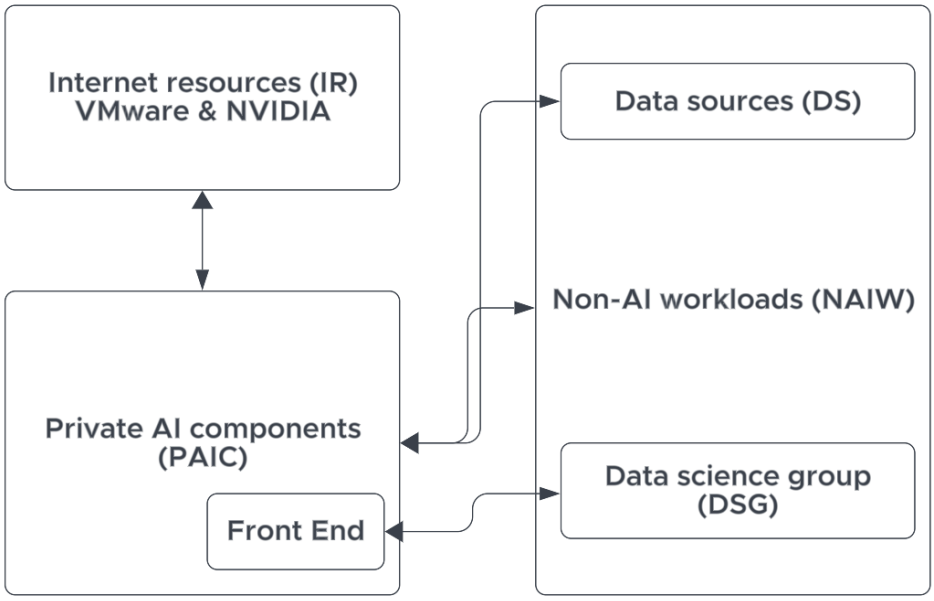
Using VMware Firewall to protect Private AI deployments

Protecting critical workloads isn't done by flipping a switch or enabling a single feature, it is a process that requires proper planning and understanding of the environment to execute a successful protection strategy. The journey of protecting a Private AI architecture deployment begins at a high level and then gets more specific as the security process progresses. At the highest level, security design begins with the following questions:

1. What destination systems do my AI components need to communicate with that are external to my private network?
2. What source systems should be able to access the Private AI components?
3. Which of the AI architecture components should be accessible to end users?

This section details the process of using the VMware Firewall to create a successful lateral control strategy for Private AI for all three of the scenarios introduced earlier: ingress, intra-component, and egress communication. Figure 12 displays a conceptual diagram of a Private AI architecture. For a more detailed look at the Private AI components, revisit Full end-to-end workflow of a RAG architecture.

Figure 12. A conceptual diagram of Private AI architecture. Each component name is shown with an associated acronym that reflects the name of the respective security group.

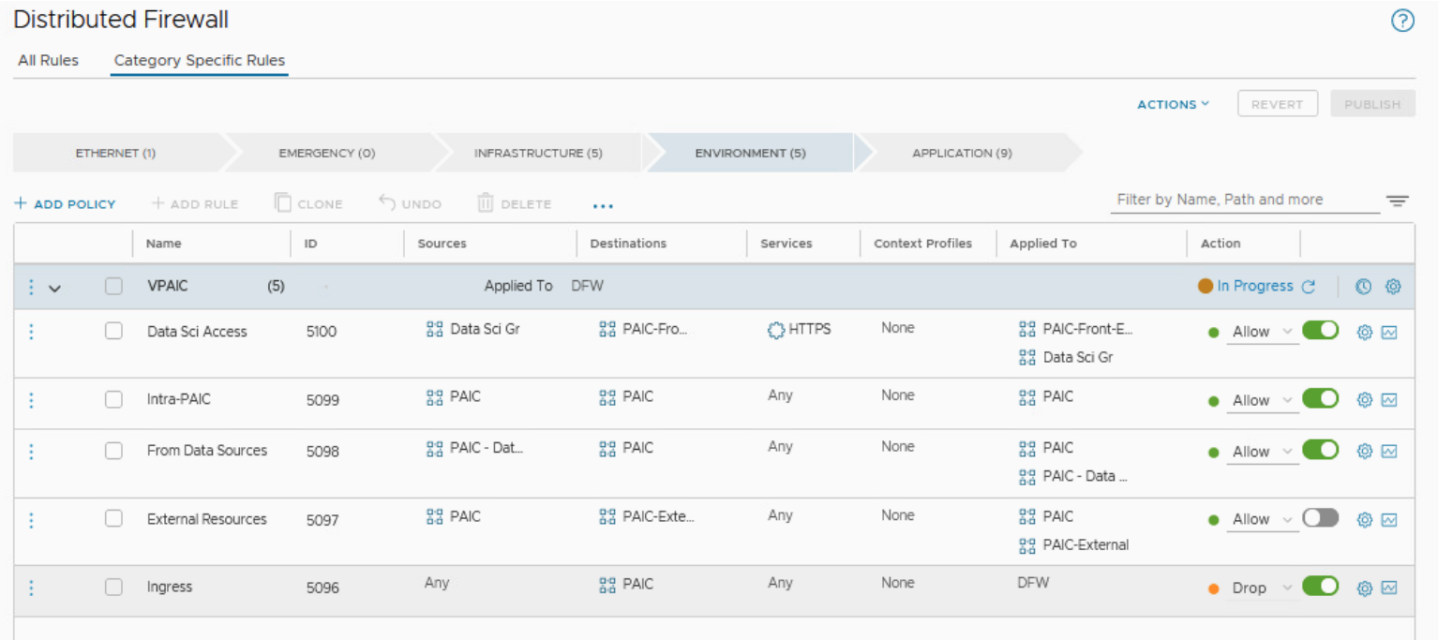


A Private AI architecture should be securely deployed by default using a defined set of rules to control lateral movement into and out of the environment. This practice ensures the integrity of the environment. The end goal is to micro-segment all of the components within the Private AI environment. The process outlined here is a multi-phased approach beginning with environmental segmentation and leads to a completely micro-segmented deployment. Each of the environmental components will be defined using NSX tags. Figure 12 shows each environmental component tag in parenthesis. As a starting point, the building blocks in this figure can be secured using the following logic:

1. All the components within the Private AI environment should be able to communicate with each other. We will dive more into this later, but for now we are going to enable all the components to talk to one another.
2. The data sources (DS) defined for the environment should be able to communicate with the Private AI components (PAIC). In a later phase, we will rebuild a more granular approach to which specific destinations are allowed (that is, the vector database).
3. The Private AI components (PAIC) should have access to a limited set of external destinations for the purpose of downloading updates. This rule is special in the sense that it does not need to be enabled all the time. As discussed earlier in this paper, the PAIC environment should be electronically air-gapped from the external world. Therefore, this rule will be disabled by default and only enabled when necessary updates require permission. Keep in mind that the rule can also be defined to limit external communications to a proxy system rather than having direct access to Internet resources. Either way, the rule should be disabled during the periods of time when the PAIC group members are not in need of updates from the Internet.
4. The data science group (DSG) needs HTTPS access to the Private AI front end to be able to test the on-going training of the learning model.
5. Communications from any other non-AI workloads should be dropped. In the beginning stages of the Private AI deployment, users do not need access to any portion of the environment until the learning model is trained. At the time the data scientist team feels the learning model is adequately trained, the end user community, in its entirety (Any) or a subset (via Security Group) can be granted access to the Private AI component that runs the chat bot or other applications that present AI data results.

Figure 13 depicts the VMware Firewall configuration of an initial security posture for the Private AI architecture. Note that these rules are created using a combination of security groups with group membership defined through NSX tags. NSX tags are different from vSphere tags. NSX tags are created within the NSX environment and do not conflict or overlap with vSphere tags. VMware vSphere tags cannot be used as the basis for security group memberships. This purposeful distinction provides a more robust, role-based access control system to prevent an unknowing vSphere administrator from mistakenly modifying the security posture of a workload by manipulating the underlying tag.

Figure 13. VMware Distributed Firewall rules implemented to provide a secure-by-default Private AI architecture



The firewall rules presented so far create a superior security posture for the Private AI environment. However, there are still additional steps to take to further increase the security of the environment. As mentioned, the end goal is to create a fully micro-segmented security configuration that controls lateral movement into, out of, and between the Private AI components. A fully micro-segmented Private AI environment will include security rules that are specific down to the port, protocol, and even layer 7 application ID respective to each component. Except for one rule, the rules shown in figure 13 are using "Any" for the Services configuration. The detailed discovery process using VMware vDefend Security Intelligence will reveal more specific information to define for the Services section of each rule.

VMware recommends two methods to reach a fully micro-segmented environment within the Private AI components:

- Use available documentation to identify the required source, destination, port, and protocol for a given communication, or…
- Use VMware vDefend Security Intelligence to monitor the Private AI environment and provide information on component flows.

VMware vDefend Security Intelligence provides the additional capability of converting the captured flow data into security rules. It's an application for flow data collection and rule recommendation, and it's a powerful tool for micro-segmentation planning. Human interaction is still required to review the flow data and determine if the flow is legitimate and valid. In the context of this paper, with a newly deployed Private AI architecture, we can confidently assume that the environment has not been compromised and therefore flow data is valid. If, however, the environment was a brownfield deployment that lacked distributed firewall policies, there is a chance that the flow data might include illegitimate flows initiated by an attacker that already reside in the environment.

Deploying a Private AI environment offers many advantages, particularly the ability to leverage intrinsic security mechanisms to protect the environment. VMware Distributed Firewall brings the concepts of a physical firewall into the hypervisor and offers vNIC-level security policy enforcement for every virtualized workload. The distributed nature of this firewall prevents hairpinning network traffic to a traditional firewall and eliminates the need to make any changes to the physical network infrastructure or the workload configuration.

Creating a secure application architecture remains a systematic process that requires strategic planning. The implementation of strong security controls impacts the environment because nothing more secure is ever easier to manage than the original environment. With VMware Distributed Firewall, the environment can be more secure without sacrificing the ease with which the environment is managed as there are no third-party tools, no agents, and no additional management interfaces. The ease and power of the distributed firewall lies in the embedded nature of the architecture and its rich contextual awareness of the applications it protects.

Conclusion

As AI development progresses beyond centralized cloud platforms and into private environments, a new set of considerations arise regarding privacy and security.

This paper explores the opportunities and potential risks associated with private AI deployments, where data remains on-premises and under user control. Privacy-preserving techniques enable powerful AI functionalities while safeguarding user information. By understanding the privacy and security implications of Private AI and implementing best practices to mitigate risks, organizations can make informed decisions about deploying these solutions and harness the transformative power of AI while maintaining user trust and regulatory compliance.

Although we identify security vulnerabilities specific to the Private AI infrastructure throughout this paper, we highlight the importance of robust access control measures and strategies to mitigate data leakage risks. For optimal security and privacy, consider implementing recommendations offered in the Open Worldwide Application Security Project (OWASP) Top 10 for LLM Applications paper and the NIST AI Risk Management Framework.

References

1. Open Worldwide Application Security Project Top 10 for LLM Applications
2. Llama 2 Responsible Use Guide
3. VMware Product Security
4. vSAN Encryption Services
5. Common Criteria Security Certification
6. Cryptographic Module Validation Program
7. Stuxnet

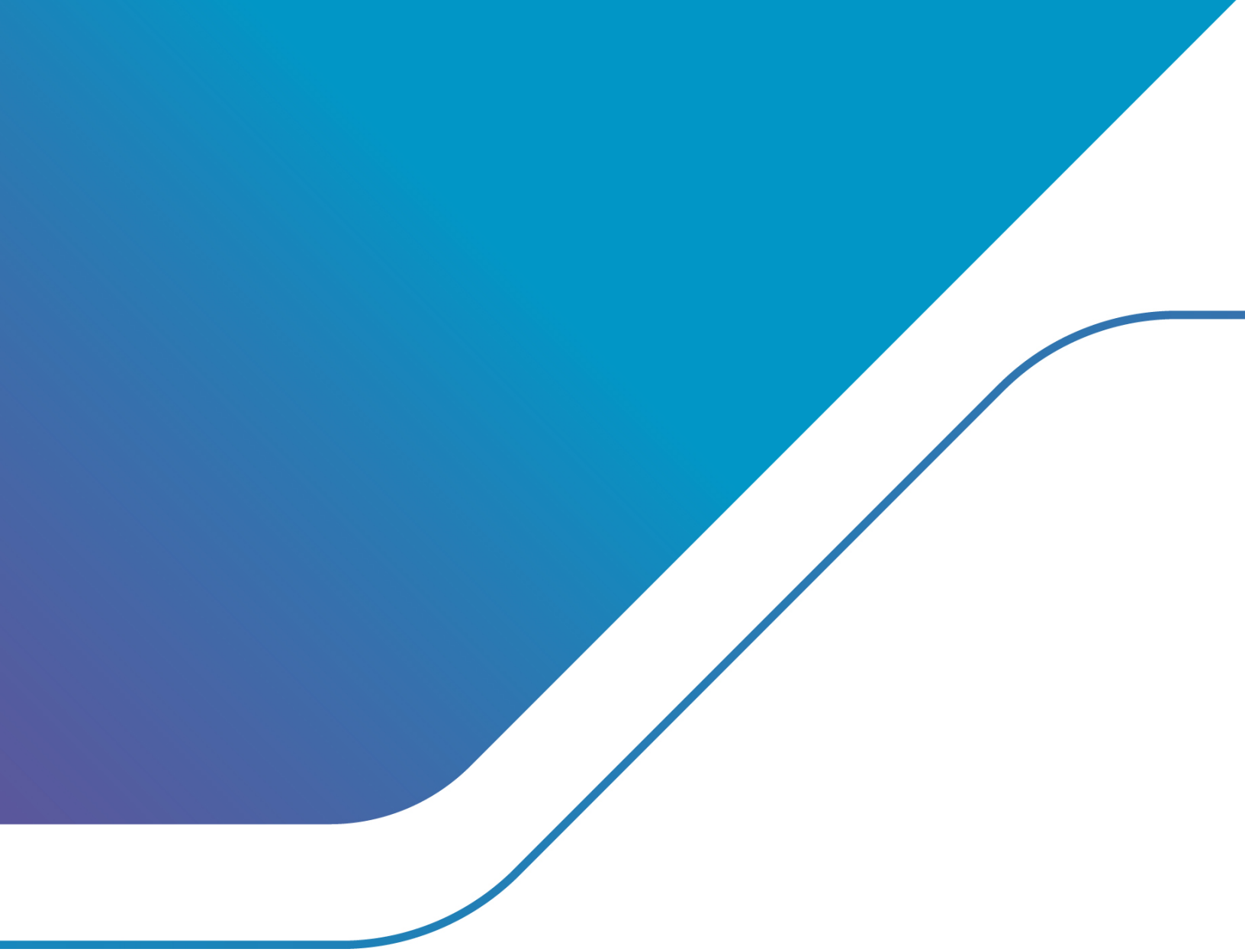
About the author

Frank Denneman serves as the chief technologist for AI at Broadcom's VMware Cloud Foundation (VCF) business unit, where he concentrates on optimizing the VMware product suite to seamlessly support modern workloads and hardware on the vSphere platform. He regularly interacts with customers and presents at top-tier events, gathering valuable feedback from customers and partners to ensure VCF products align with their requirements. He holds the VMware Certified Design Expert certification, has authored the books "vSphere Clustering and Resource Deep Dives," and manages the website frankdenneman.nl. Additionally, Frank hosts the podcast "Unexplored Territory."

Chris McCain is a product management director in the Application Networking and Security franchise at VMware by Broadcom. Chris has over 20 years of experience in the world of information technology. In addition to being a double VCDX in data center virtualization and network virtualization, Chris has written several books on virtualization and infrastructure design. As part of his mission at VMware, Chris travels around the world talking to customers and partners about VMware vision and strategy for next generation enterprise architecture. Chris is based in St Petersburg, FL.

Acknowledgments

We would especially like to thank Jenny Risdal, Bhavani Kumar, Sujata Banerjee, Bob Plankers, Fanny Strudel, Chris Gully, Shawn Kelly, Agustin Malanco Leyva, Ramesh Radhakrishnan, and Chris Wolf for their help and support in completing this work.



vmware® by Broadcom