Bringing Large Language Models Inside the Corporate Firewall
A Technical White Paper for CIOs and Chief AI Architects

---

1. Executive Overview

Enterprises want the productivity and intelligence of large language models (LLMs) without exposing proprietary data to the public internet or to third-party training pipelines. The good news: this is not only possible, it is now a well-trodden path.

There are three primary patterns for keeping LLM workloads inside your trust boundary:

1. **Private connectivity to a managed LLM service** (Azure OpenAI, AWS Bedrock, Google Vertex AI) using VNet/VPC isolation and private endpoints—no public internet exposure, no training on your data. Google Cloud Documentation+4Microsoft Learn+4AWS Documentation+4
2. **Self-hosted LLMs in your private cloud** (your own VPC/subscription/tenant) on managed Kubernetes or VM clusters, using open-source or licensed models. TrueFoundry+2allganize.ai+2
3. **Fully on-premises LLMs** in your data center, running on your GPU servers with no external network path at all. ACL Anthology+2aiveda.io+2

In all three models, you can design an architecture where:

- **Prompts, documents, and embeddings never traverse the public internet**
- **Providers do not use your data to train or improve foundation models**
- **All data at rest and in transit is encrypted with keys you control**
- **Access is governed by your IAM / identity provider, with full auditability**

This paper explains how to achieve that in technical depth and gives you a reference architecture you can take into a design review, security committee, or board meeting.

---

2. Deployment Models: Keeping the LLM Inside Your Trust Boundary

2.1 Managed LLM, Private Network (Azure, AWS, GCP)

Modern hyperscaler offerings explicitly support “inside the corporate firewall” semantics using private networking, zero-data-retention, and tenant-level isolation.

Azure OpenAI

- Azure OpenAI runs in your Azure subscription and tenant; prompts and completions are processed within Microsoft's cloud and are **not used to train OpenAI's public models**. ais.com+4Microsoft Learn+4Microsoft Azure+4
- You can integrate the service into a **private VNet** and expose it only through **private endpoints** and Azure Private Link, preventing access from the public internet. basofa.com+3Microsoft Learn+3Microsoft Learn+3
- "On your data" / RAG features connect to Azure Storage, Cosmos DB, or Azure AI Search; the data **remains in your designated data stores and region**, and the model never persists your documents. Microsoft Learn

AWS Bedrock

- Amazon Bedrock provides access to multiple foundation models (Anthropic, Meta, Amazon Titan, etc.) and commits that **your data is not used to train the underlying models** by default. AWS Documentation
- You can access Bedrock via **VPC interface endpoints** using AWS PrivateLink so that LLM calls occur over the **AWS backbone only**, with no internet gateway, NAT, or public IPs. GitHub+5AWS Documentation+5AWS Documentation+5

Google Cloud Vertex AI

- Vertex AI's generative services operate under Google's AI/ML Privacy Commitment: **customer prompts and data are not used to train foundation models** and can be constrained to specific data residency regions. Google Cloud+5Google Cloud Documentation+5Google Cloud Documentation+5
- You can restrict access using VPC Service Controls, private service connect, and organization policies, keeping LLM calls inside your private network perimeter.

Security Takeaway:
With proper configuration, these services behave like an internal microservice inside your network: reachable only via private endpoints, governed by your IAM, with zero data retention for training and fully auditable logs. From a CISO's perspective, they often look similar to any other PaaS database or analytics service you already consume from a hyperscaler.

2.2 Self-Hosted LLMs in a Private Cloud (Your Own VPC/Tenant)

If you need even more control (custom models, non-standard architectures, or specific sovereignty rules), you can host LLMs yourself inside your own private cloud.

Key building blocks: Database Mart+5TrueFoundry+5allganize.ai+5

- **Compute layer**: GPU-enabled nodes (e.g., NVIDIA A10/A100/H100) in your Kubernetes cluster or VM scale set.
- **Model server**: frameworks like vLLM, Text Generation Inference, llama.cpp, or vendor toolkits (NVIDIA NIM, OnPrem.LLM, etc.) to host models like Llama 3, Mistral, or licensed enterprise models.
- **Private networking**: subnets, security groups/NSGs, and firewall rules that expose the model only to internal application tiers or API gateways.
- **Storage & features**:
    - Encrypted object or block storage for model checkpoints.
    - Encrypted vector databases (e.g., pgvector, Milvus, Pinecone with private endpoints) for RAG.
    - Internal feature stores or document stores for grounding data.

In this model, **no LLM traffic leaves your cloud account**:

- Model weights live in your own storage, in your region of choice.
- All inference happens on compute nodes in your tenant.
- You can disable all outbound internet access at the subnet or firewall level.

This is functionally equivalent to any other line-of-business service you host in your VPC today, with the added GPU footprint and model-specific observability.

2.3 Fully On-Premises LLMs

For highly regulated workloads (defense, critical infrastructure, certain healthcare or financial institutions), you may need LLMs entirely in your own data centers.

Recent work and tooling emphasize **on-prem LLM deployment** specifically for privacy-sensitive environments: IKANGAI+4TrueFoundry+4ACL Anthology+4

- Deploy GPU clusters (bare metal or virtualized) behind your corporate firewalls.
- Run a Kubernetes cluster or similar orchestration platform.
- Host LLMs with open-source or contracted commercial licenses.
- Optionally use confidential computing (TEE-enabled CPUs/GPUs) to add hardware-level isolation.

In this configuration:

- There is **literally no network path from the LLM stack to the internet** except what you explicitly allow.
- All packets stay on your corporate backbone; all data at rest is on your storage arrays.

- Security posture can be validated using the same tooling you use today (NAC, IDS/IPS, DLP appliances, SIEM).

---

3. Reference Architecture: LLM Inside the Corporate Firewall

Below is a logical architecture that applies to all three deployment options, with implementation details varying by environment.

3.1 Network Isolation and Perimeter Controls

Objective: Ensure prompts and documents never traverse the public internet and cannot be accessed by other tenants.

Key patterns:

1. **Private subnets & segmentation**
    - Place LLM inference services in dedicated subnets (e.g., “ai-core”) with:
        - No public IPs
        - No direct outbound internet access
        - Only east-west traffic allowed from a limited set of service subnets (API gateways, app backends).
2. **Private endpoints / service endpoints**
    - For managed LLM services: expose them via **VNet/VPC private endpoints** (Azure Private Link, AWS PrivateLink, GCP Private Service Connect). Amazon Web Services, Inc.+6AWS Documentation+6Microsoft Learn+6
    - Enforce that traffic to the LLM service is possible only from within corporate networks/VPN.
3. **Egress control**
    - Use NAT, explicit proxies, or egress gateways for any permitted outbound connections; default is deny-all.
    - Implement L7 egress policies (e.g., Kubernetes network policies + service mesh) to restrict which services can talk to what external hosts.
4. **Zero-trust service mesh (optional but recommended)**
    - Use a service mesh (Istio, Linkerd, etc.) with:
        - Mutual TLS between microservices
        - Policy-based access control (“only these services may call /v1/chat/completions”)
        - Distributed tracing for auditing and incident response.

Result: even if a developer misconfigures an app, it **cannot accidentally send prompts to an external SaaS LLM**; the only reachable LLM endpoints are those inside your network or private cloud.

3.2 Identity, Access, and Secrets Management

Objective: Only authorized services and users can interact with LLMs and underlying data.

Patterns:

- **Central IdP**: Use enterprise identity (Entra ID, Okta, Ping, etc.) for:
  - Admin access to LLM platforms and portals.
  - Human tooling (e.g., internal chat interfaces) with role-based entitlements.
- **Service-to-service auth**:
  - Use workload identity (Kubernetes service accounts bound to IAM roles, managed identities, etc.) instead of embedded keys.
  - Grant least-privilege permissions: a given microservice can call the LLM but not access its model weights directly.
- **Secrets management**:
  - Store API keys, model route configuration, and encryption keys in a central vault (Azure Key Vault, AWS KMS/Secrets Manager, HashiCorp Vault).
  - Rotate keys automatically; log all secret access.
- **Fine-grained RBAC**:
  - Distinguish “Prompt Designer,” “Data Engineer,” “Model Ops,” “Security Admin,” etc., with explicit scopes for access to logs, prompts, vector indices, and telemetry.

This gives you clear control over *who* can cause data to flow into the LLM and *what* they can see in return.

3.3 Data Plane: RAG, Vector Stores, and Data Residency

In secure enterprise deployments, you typically **do not upload your corpus into the model itself**. Instead, you keep data in your existing stores and use Retrieval-Augmented Generation (RAG):

1. **Data stays in your systems**
   - Documents live in your DMS, object stores, or databases—inside your data center or private cloud.
   - For managed RAG (“on your data” in Azure OpenAI, Vertex AI Search, Bedrock knowledge bases), the provider explicitly states that data remains in your selected storage and region. Nerdio Manager+4Microsoft Learn+4Google Cloud Documentation+4

2. **Embeddings & vector stores**
    - You compute embeddings either:
        - Locally (self-hosted embedding model), or
        - Via a managed embedding service accessed through the same private networking constructs.
    - Store embeddings in an **encrypted** vector database:
        - At rest encryption with KMS-managed keys.
        - Row-level or tenant-level security for multi-business-unit scenarios.
3. **RAG flow**
    - Application receives user query.
    - Query is passed to a retriever, which:
        - Performs semantic search across the vector store.
        - Applies document-level access control ("only docs this user or system is allowed to see").
    - Only **relevant snippets**, not full documents, are concatenated to the prompt and sent to the LLM.
4. **No long-term storage in the LLM**
    - The model is stateless with respect to your request; after inference, memory is freed.
    - For logging, you can:
        - Redact or hash sensitive fields before writing logs.
        - Apply retention policies and DLP scanning on logs.

Because data never leaves your storage plane—and the model never retains it—the privacy risk profile is much closer to "read-only analytics" than "uploading the dataset to a third-party SaaS."

3.4 Model Lifecycle: Training, Fine-Tuning, and Isolation

If you need fine-tuned or domain-specialized models, there are secure patterns that keep training data private.

1. **Managed fine-tuning with zero data retention**
    - Hyperscalers offer fine-tuning services where training data is used only to create your private model variant and is not used to update the shared base model; the tuned model is available only in your tenant. Medium+4Microsoft Learn+4Google Cloud Documentation+4
    - Combine this with private networking and encryption to ensure training jobs access data only via private endpoints.
2. **Self-hosted training**
    - For highly sensitive datasets, run the entire training pipeline on your own compute (private cloud or on-prem).
    - Use:

            - Encrypted storage for checkpoints.
            - Isolated training clusters (no mixed-tenant workloads).
            - Regular vulnerability scans and hardening for driver/OS layers (GPU drivers, CUDA runtime, etc.).
3. **Confidential computing (optional)**
    - Some platforms support **confidential VMs** and **TEEs** that encrypt data in use and provide attestation that code and data cannot be accessed by the cloud provider or other tenants, adding a hardware-backed assurance layer. Google+1

The net effect: whether you fine-tune in a cloud tenant or entirely on-prem, you can guarantee that training data and resulting weights remain within your trust boundary and are not co-mingled with public models.

3.5 Observability, DLP, and Governance

Keeping data private is not only about architecture; you need continuous assurance.

- **Centralized logging & SIEM**
    - Ship LLM request/response logs (with appropriate redactions) to your SIEM.
    - Define detections for anomalous access patterns, unusual volume, or unexpected prompts involving secrets.
- **DLP integration**
    - Apply DLP policies at:
        - The user interface layer (prevent users from pasting SSNs, credit card numbers, etc.).
        - The network layer (DLP-enabled proxies for any residual egress).
        - The log/analytics layer (scan logs for sensitive patterns).
- **Policy & governance**
    - Define allowed and prohibited use cases for LLMs.
    - Establish an AI Governance Council to review new use cases, model changes, and high-risk data domains.
- **Model risk management**
    - Treat LLMs as high-risk models in your MRM framework:
        - Document training data provenance and controls.
        - Maintain versioned model inventories.
        - Conduct periodic red team tests (prompt injection, data exfiltration, jailbreak attempts).

This governance layer is what “proves” to internal audit and regulators that your architectural controls are not just designed but actually enforced and monitored.

4. Proving Data Privacy: Control Objectives and How They're Met

For CIOs and Chief AI Architects, conversations with CISOs, auditors, and regulators typically boil down to concrete assurances. Below are key control questions and how the described architectures answer them.

4.1 Does Proprietary Data Leave Our Trust Boundary?

- **Managed LLM with private endpoints**
  - Data flows from your VNet/VPC to the provider's service over private network links.
  - Providers explicitly commit that:
    - **Prompts and completions are not used to train or improve shared foundation models.**
    - **Data is not shared across tenants or with public services such as chat.openai.com.** ais.com+7Microsoft Learn+7Google Cloud Documentation+7
  - Data does leave your VNet/VPC but remains within the provider's regional infrastructure, under contractual and regulatory controls comparable to your other cloud workloads (databases, analytics).
- **Self-hosted or on-prem LLM**
  - No traffic leaves your cloud account or data center.
  - All flows are confined to networks and storage that you own and operate.

Answer: For private cloud or on-prem deployments, data never leaves your trust boundary by design. For managed LLMs with private networking, data stays within the provider's infrastructure but is not exposed to the public internet or used outside your tenant.

4.2 Is Our Data Used to Train Other Customers' Models?

All major enterprise-focused generative AI platforms now offer **zero-data-retention**/no-training commitments for customer data:

- Azure OpenAI, Vertex AI, and Bedrock explicitly state that customer content is not used to train foundation models (unless you opt into specific programs). Google Cloud+6Microsoft Learn+6Google Cloud Documentation+6
- Self-hosted and on-prem scenarios inherently avoid cross-tenant training.

Answer: With the correct configuration and contracts, your data is *not* used to train other customers' models or public services.

4.3 Can We Prove Encryption and Access Controls?

- In all three deployment modes, you can require:
    - **Transport encryption** (TLS 1.2+ end-to-end).
    - **At-rest encryption** with keys managed by your own KMS or HSM.
    - **Fine-grained IAM policies** that log every call and admin action.
- Hyperscalers' gen-AI services piggyback on their existing security posture (certifications, key management, network isolation). ais.com+5AWS Documentation+5Google+5

Answer: Yes. You can demonstrate encryption, identity, and logging configurations as you already do for other critical systems (ERP, clinical systems, financial ledgers).

4.4 How Do We Prevent Accidental Leakage to Public LLMs?

- **Technical controls**
    - Egress firewall rules and proxy policies that block access to known public LLM endpoints (e.g., *.openai.com, *.anthropic.com) except from sanctioned "labs" segments.
    - Browser isolation or CASB policies that restrict uploading corporate content to generative AI websites.
- **Process controls**
    - Corporate AI policy that channels employees to internal LLMs.
    - Training and regular reminders.

Answer: By combining network-level blocks, CASB/DLP, and policy, you can materially reduce the risk of employees leaking IP to public tools and steer usage to controlled internal LLMs.

---

5. Implementation Blueprint

A pragmatic roadmap for bringing LLMs inside your firewall:

Phase 0 – Strategy & Guardrails

- Define:

- High-value use cases (e.g., developer assist, knowledge search, customer service).
- Data classification rules: which data can be used, under which conditions.
- Target deployment model(s): managed private LLM vs self-hosted vs on-prem, aligned to risk and regulatory constraints.

Phase 1 – Secure Pilot (Single Use Case)

- Spin up a **sandbox VNet/VPC** or on-prem segment with:
  - Private endpoints to an enterprise LLM service *or* a small self-hosted model deployment.
  - Locked-down egress and IAM.
- Implement:
  - A single RAG-based application using a small, **non-sensitive** document set.
  - Full logging, DLP at UI, and clear user messaging.
- Perform:
  - Security review (architecture, IAM, logging).
  - Red-team exercise focused on prompt-injection and data exfiltration.

Phase 2 – Core Platform Build-Out

- Stand up a **shared AI platform** with:
  - Central model gateway (API) that routes to various models (managed and/or self-hosted).
  - Shared vector store and RAG services with tenancy and access control.
  - Standard patterns for:
    - Secret management.
    - Request/response logging and redaction.
    - Prompt templates and guardrails.
- Integrate:
  - Enterprise identity.
  - SIEM/SOAR for alerting and incident response.
  - Change management and release pipelines for model updates.

Phase 3 – Scale to Multiple Use Cases

- Onboard priority functions (e.g., IT, Legal, Finance, Operations) with clearly scoped RAG corpora.
- For each use case:
  - Conduct a data protection impact assessment (if required).
  - Configure document-level permissions in the retrieval layer.
  - Measure value (productivity, response quality) and adjust.

Phase 4 – Advanced Privacy & Sovereignty

- For the most sensitive domains:
  - Consider **on-prem or sovereign cloud** deployments with no external dependency.
  - Use confidential computing where available for training and inference.
  - Implement differential privacy or other advanced techniques for analytics on model behavior, if needed. Mithril Security Blog+3ACL Anthology+3Google+3

---

6. Conclusion

You do not have to choose between “use LLMs” and “protect proprietary data.” The current ecosystem supports robust patterns to bring LLM capabilities **inside your corporate firewall**, whether that firewall surrounds a private VPC, a dedicated cloud tenant, or your own data center.

By combining:

- Private networking (VNet, VPC, private endpoints),
- Strong identity and key management,
- RAG architectures where data stays in your stores,
- Zero-data-retention managed services or self-hosted/on-prem models,
- And disciplined logging, DLP, and governance,

you can demonstrate—technically and to auditors—that sensitive prompts and documents remain private, confidential, and under your control while still unlocking the transformational benefits of generative AI.