MaviNovo AI

Client Data Privacy & Security

Powered by the MN2 Framework

Overview

At MaviNovo AI (MNAI), safeguarding client data is not a feature — it is a foundational commitment. Every architectural decision, technology choice, and operational policy is made with privacy, security, and enterprise accountability at the forefront. This document outlines how MNAI protects client data at every layer of the platform, from the underlying MN2 Framework to its integration with external AI services.

1. Data Encryption

All client data transmitted through and stored within the MaviNovo AI platform is encrypted. Whether data is in transit or at rest, MNAI enforces encryption across all endpoints and data stores hosted on the MNAI server. This ensures that client information is never exposed in plaintext at any point in the system.

MN2 supports standard HTTPS configurations natively, securing all communication between clients and the platform. This is not bolted on after the fact — it is built into the framework's communication architecture from the ground up.

2. The MN2 Framework Security Foundation

MNAI is powered by the MN2 Framework — a proprietary, enterprise-grade platform built in Java over 12 years of real-world enterprise software experience. Security is not an afterthought in MN2; it is embedded in every design decision.

2.1 Pure Java Architecture

MN2 is written entirely in Java — a mature, hardened language with a long track record in enterprise security. Unlike platforms built on JavaScript or Python ecosystems, which are frequently targeted by dependency chain attacks and global zero-day vulnerabilities, MN2 maintains an extremely small attack surface. This disciplined, minimal-dependency design means MNAI is immune to many classes of vulnerabilities that have crippled other platforms, including the infamous Log4j exploit that affected a vast portion of the software industry.

2.2 Enterprise Auditing & Traceability

One of MN2's most powerful security features is its built-in enterprise auditing. Most user action within the system is tracked, and every database field retains its full history. This creates a complete and transparent chain of custody for client data — something most modern AI platforms do not even attempt to provide.

This means clients can audit much of the data related to who accessed data, what changed, and when — providing the accountability that regulated industries and enterprise environments demand.

2.3 Minimal Attack Surface & Dependency Control

Modern software vulnerabilities often originate not in the application itself, but in its dependencies. MN2 was engineered with a philosophy of minimal dependencies — meaning fewer third-party libraries, fewer potential vulnerabilities, and a smaller surface for attackers to exploit. Every dependency included in MN2 has been deliberately evaluated, reducing supply chain risk significantly.

2.4 Flexible & Secure Deployment

MN2's lightweight, self-contained architecture enables deployment anywhere — cloud, on-premises, or private network — with nothing more than SSH access. This means enterprise clients can keep MNAI entirely within their own controlled infrastructure if required, with no dependency on third-party cloud providers like AWS or Azure unless desired.

MN2 runs securely behind firewalls and supports standard HTTPS and SMTP configurations. Clients retain full control over their infrastructure and data — with no vendor lock-in and no hidden dependencies.

3. LLM Integration & Data Retention Policies

MNAI supports integration with both external Large Language Models (LLMs) and local, self-hosted LLMs. The privacy and data retention implications differ between these two approaches, and clients can choose the model that best suits their needs.

3.1 External LLMs with Zero Data Retention

When clients choose to integrate MNAI with a qualifying external LLM provider — such as OpenAI — they can take advantage of zero data retention API policies. Under these policies, data submitted via the API is not stored, logged, or used for model training or improvement by the LLM provider. This means:

- Client data sent to the LLM during AI processing is not retained by the provider.
- Client data is not used to train or fine-tune the external model.
- Each API interaction is treated as ephemeral — processed and discarded.

MNAI strongly recommends that enterprise clients configure their LLM integrations using API-based access with zero data retention policies wherever available, to ensure maximum data privacy.

3.2 Local LLMs via MNAI Vector Database

For clients requiring complete data sovereignty — where no data can leave their environment under any circumstances — MNAI includes a built-in vector database that enables the use of locally hosted LLMs. In this configuration, all AI processing happens entirely within the client's own infrastructure. No data is ever transmitted to an external service.

This approach is ideal for organizations operating in highly regulated industries or sensitive environments. However, it is important to understand the trade-off: local LLMs are typically less capable than leading external models such as OpenAI or Anthropic. They may produce lower-quality outputs, handle complex reasoning less effectively, and have more limited context windows.

MNAI's vector database provides the infrastructure foundation for local AI — enabling semantic search, retrieval-augmented generation (RAG), and context-aware processing — all without any external data exposure.

3.3 LLM Approach Comparison

Factor	External LLM (e.g. OpenAI)	Local LLM (MNAI Vector DB)
Data Leaves Environment	Yes (via API, no retention with ZDR policy)	No — fully local
Used for Model Training	No (with zero data retention API policy)	No
Model Capability	High — leading commercial models	Moderate — varies by model
Data Sovereignty	Partial — depends on provider policy	Complete
Deployment Complexity	Low	Higher — requires local infrastructure
Best For	Maximum AI capability with strong privacy controls	Highly regulated or classified environments

4. Infrastructure Control & No Vendor Lock-In

MNAI is designed to give clients complete control over where and how their data lives. Because MN2 can deploy to any environment — cloud, on-premises, or private network — organizations are never dependent on a single cloud vendor. There are no mandatory integrations with AWS, Azure, or Google Cloud, and clients can migrate or self-host at any time.

This architecture also means that the performance and security characteristics of the platform are consistent regardless of deployment environment. By pairing Tomcat's CPU and network performance with MySQL's disk efficiency on a single server, MN2 eliminates the internal latency introduced by distributed architectures — while keeping the security perimeter clean and manageable.

5. Backend / Frontend Separation

MN2 enforces a clean architectural separation between the backend and frontend, using JSON-based AJAX services for all communication. This design means that the backend logic — where data is processed, stored, and secured — is never directly exposed to the client layer. Updates to user interfaces or the addition of new modules do not require changes to the backend, and vice versa.

This separation reduces the risk of data leakage through the presentation layer and ensures that security controls remain centralized and consistently enforced, regardless of how clients interact with the platform.

6. Summary of Security Capabilities

Capability	MNAI Approach
Data Encryption	All data encrypted in transit and at rest across mavinovoai.com
Audit Trail	Every user action and database field change tracked and retained
Language Security	Built in Java — minimal dependencies, small attack surface
LLM Data Privacy	Zero data retention API policies available with qualifying providers
Local AI Option	Built-in vector database supports fully local LLM deployments
Deployment Flexibility	Deploy to cloud, on-premises, or private network via SSH
Firewall Compatible	Runs securely behind enterprise firewalls
No Vendor Lock-In	No mandatory dependency on AWS, Azure, or any cloud provider
Backend Isolation	Frontend/backend separation via JSON-based AJAX services

MaviNovo AI was built for enterprises where security and accountability are non-negotiable. From the Java core of MN2 to its deployment flexibility and LLM privacy controls, every layer of the platform has been engineered to protect what matters most — your clients' data.

