NM 3.0 Technology Stack Overview

This document outlines the proposed technology stack for the NM 3.0 application, focusing on delivering a modern, scalable, and secure solution that aligns with current industry best practices. The selection of tools and frameworks has been carefully made to support rapid development, efficient deployment, and long-term maintainability. By leveraging cutting-edge technologies across the frontend, backend, database, and DevOps layers, the goal is to build a robust architecture capable of handling evolving business needs and high user demand.

Application	.NET Core	Web Server	NGINX, Apache Web Server
DMS	LogicalDOC	CMS	umbraco
DataBase	PostgreSQL	Mobile App	Flutter
Container	docker	Orchestration	kubernetes
API Gateway	Ocelot	Circuit Breaker	Polly
Message Queue	RabbitMQ	Service Discovery	Consul
Analytics & Visualization	Smart Pulse	Version Control	git
Log Analytics	ELK	IDAM & SSO	KEYCLOAK

NM 3.0 Technology Stack Overview

Application Framework

.NET (Cross-platform)
A high-performance, open-source framework for building modern, cloud-enabled, and internet-connected applications, including APIs, web apps, and microservices.

Web Servers

NGINX
A high-performance HTTP server and reverse proxy, suitable for serving static content and enabling load balancing.

Apache
A reliable, open-source web server with support for dynamic modules, including integration with PHP and other scripting languages.

Content Management

LogicalDOC
An open-source document management system for storing, managing, and tracking electronic documents and images.

Umbraco CMS
An extensible, .NET-based open-source CMS used for building and managing content-rich digital experiences.

Database

PostgreSQL
A powerful, open-source object-relational database system known for its reliability, performance, and extensibility.

UI Toolkit

Flutter
An open-source UI toolkit developed by Google for building natively compiled applications across mobile, web, and desktop using a single codebase.

Containerization and Orchestration

Docker
An open-source platform for building, packaging, and running applications in isolated containers.

Kubernetes
An open-source system for automating the deployment, scaling, and management of containerized applications.

Microservices Architecture Components

Ocelot
A lightweight .NET API Gateway supporting routing, request aggregation, and security for microservices-based applications.

NM 3.0 Technology Stack Overview

Polly
A .NET resilience and transient-fault-handling library that provides retry, circuit breaker, timeout, and fallback policies.

RabbitMQ
An open-source message broker that facilitates asynchronous communication between distributed systems.

Consul
A service mesh solution offering service discovery, configuration, health checking, and segmentation.

Monitoring and Observability

Custom Monitoring Tools
Supports real-time monitoring and performance visualization for key application metrics.

ELK Stack (Elasticsearch, Logstash, Kibana)
Used for log aggregation, transformation, and visualization to support centralized logging and troubleshooting.

Version Control and IAM

Git
A distributed version control system that enables efficient collaboration and source code management.

Keycloak
An open-source Identity and Access Management (IAM) solution that provides single sign-on (SSO), user federation, LDAP integration, and multi-factor authentication (MFA).

Data Migration Plan

This section outlines the strategy to migrate data from Microsoft SQL Server to PostgreSQL with minimal downtime and data integrity assurance.

1. Primary Tool: pgLoader

Description:
pgLoader is an open-source command-line tool designed for migrating data from various databases to PostgreSQL.

Usage:

- Schema conversion (tables, indexes, constraints)
- Data migration (parallelized and optimized)

Benefits:

- Automates schema transformation
- High-performance parallel loading
- Built-in error handling and retries

2. Custom Migration Scripts

When to Use:
For scenarios requiring custom transformations, selective migrations, or adaptation of business-specific logic.

Use Cases:

- Data cleansing or restructuring
- Manual conversion of stored procedures and triggers
- Partial table/column-level migration

Tools:

- SQL Server Management Studio (SSMS)
- PostgreSQL tools (psql, pgAdmin)
- Python with libraries (e.g., psycopg2, pandas)

3. ETL-Based Approach (Optional/Fallback)

When to Use:
Large datasets, multiple source systems, or complex transformation rules.

Tooling:

- Custom ETL using .NET

Advantages:

- Real-time/incremental migration support
- Detailed monitoring and logging
- Visual transformation workflows

Validation & Rollback Strategy

- Post-migration validation: row counts, checksums, referential integrity
- Rollback readiness: backups and reverse scripts available for recovery

---

Security Best Practices

Authentication & Authorization

- Role-based and policy-based access control
- Integration with Keycloak for identity federation and session management
- Secure session and token handling

NM 3.0 Technology Stack Overview

Data Protection

- Data encrypted at rest and in transit
- Secrets management using environment variables or User Secrets
- No credentials stored in source code

Secure Communication

- HTTPS enforced (TLS 1.2 or 1.3)
- Secure API authentication using JWTs with token expiry

Input Validation & Threat Prevention

- Protection against:
  - SQL Injection
  - Cross-Site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
- Input sanitization and encoding best practices

Secure Development & CI/CD

- Static code analysis (SAST)
- CI/CD pipelines with secret management
- Dependency vulnerability scanning

Monitoring & Incident Response

- Real-time application logging and monitoring
- Alerting on anomalies or suspicious activities
- Periodic penetration testing and security audits

Security Commitment

- Security-first architecture design
- Transparent reporting and threat visibility
- Continuous security posture improvement

NM 3.0 Technology Stack Overview

Key Issues Resolutions in existing NM 2.0

Session Management Issue

Problem:
Users are unable to log in if they did not log out properly in a previous session.

Solution:
Integrate Keycloak to handle centralized authentication, session timeout, and forced logouts, ensuring seamless and secure session handling.

Manual Payment Reconciliation

Problem:
Manual reconciliation of payments is error-prone and time-consuming.

Solution:
Implement an Automated Payment Reconciliation module to match records accurately, reducing manual efforts and improving transaction reliability.

Monolithic Architecture Performance Bottlenecks

Problem:
Current monolithic application structure causes performance issues and limits scalability.

Solution:
Refactor into a microservices architecture for modularity, improved performance, and independent scalability.

Inefficient Document Management

Problem:
Storing documents in the database increases size and affects performance.

Solution:
Introduce LogicalDOC as a dedicated document management system to offload storage and enhance maintainability.

Status Mismatch and Service Downtime

Problem:
Lack of message queuing causes request loss during service unavailability.

Solution:
Integrate RabbitMQ for asynchronous communication and message durability, ensuring system resilience.

NM 3.0 Technology Stack Overview

Manual Deployment Downtime

Problem:
System is offline during manual deployments, impacting availability.

Solution:
Adopt a CI/CD pipeline to automate deployments with zero downtime, enabling continuous delivery and high availability.