Designing a Hybrid ORM Architecture for High-Performance .NET Applications

Corresponding Author: Mehrdad Azimi, **Email**: AzimiDeveloper@Gmail.com - 2025

Introduction

In modern .NET software development, using Object-Relational Mapping (ORM) tools is standard practice for database access. However, choosing between heavyweight ORMs like Entity Framework Core and lightweight alternatives like RepoDb always presents a trade-off between performance and features.

This paper introduces a hybrid approach that combines EF Core and RepoDb to benefit from the strengths of both systems. EF Core is used for complex relational operations, while RepoDb serves high-performance, lightweight data interactions. This article examines the architecture, implementation structure, performance benchmarks, and comparative benefits of this hybrid model.

Technical Comparison of EF Core and RepoDb

Feature	EF Core	RepoDb
IQueryable Support	Yes	No
CRUD Operation Speed	Relatively Slow	Very Fast
Complex Query Capabilities	Excellent	Limited
Change Tracking	Yes	No
Migration Support	Supported	Not Supported
Best Use Case	Complex Scenarios, Navigation	Simple and Fast CRUD Operations

Hybrid Architecture

The proposed architecture separates responsibilities into two repository types based on the workload:

1. EfBaseRepository<TEntity>

A generic repository based on EF Core, providing:

- Full IQueryable support
- Methods like InsertAsync, Update, Delete, SaveAsync
- Navigation loading via Include, Change Tracking, and Migration

2. RepoDbBaseRepository<TEntity>

A lightweight repository for direct and high-speed operations:

- No overhead from change tracking
- High-speed support for Select, Insert, Update, Delete
- Ideal for dashboards, lightweight listings, and background jobs

Performance Charts: Response Time and Memory Usage

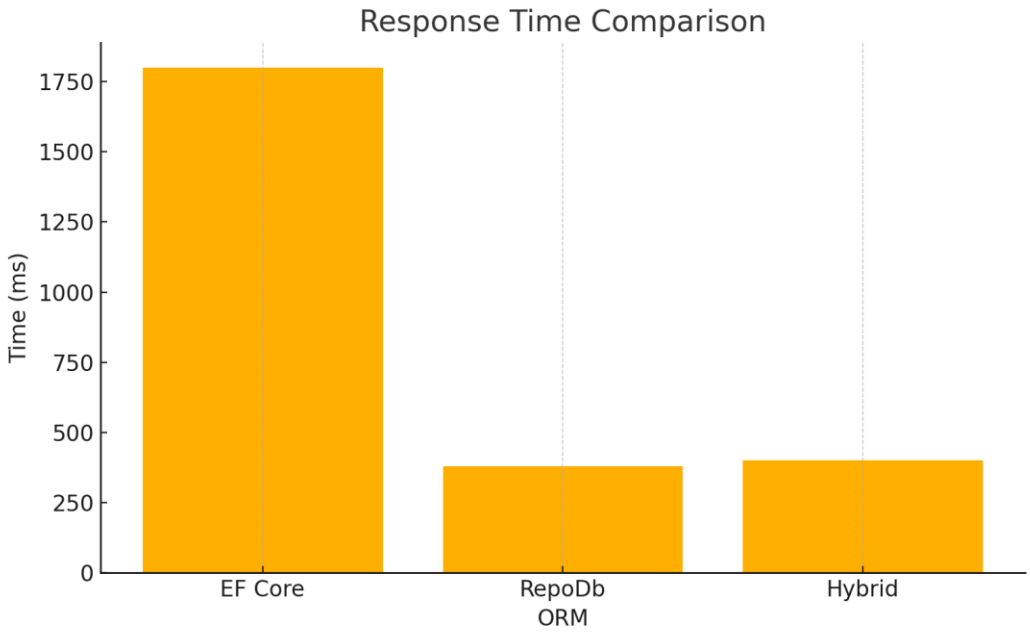

Chart: Response Time Comparison (ms)

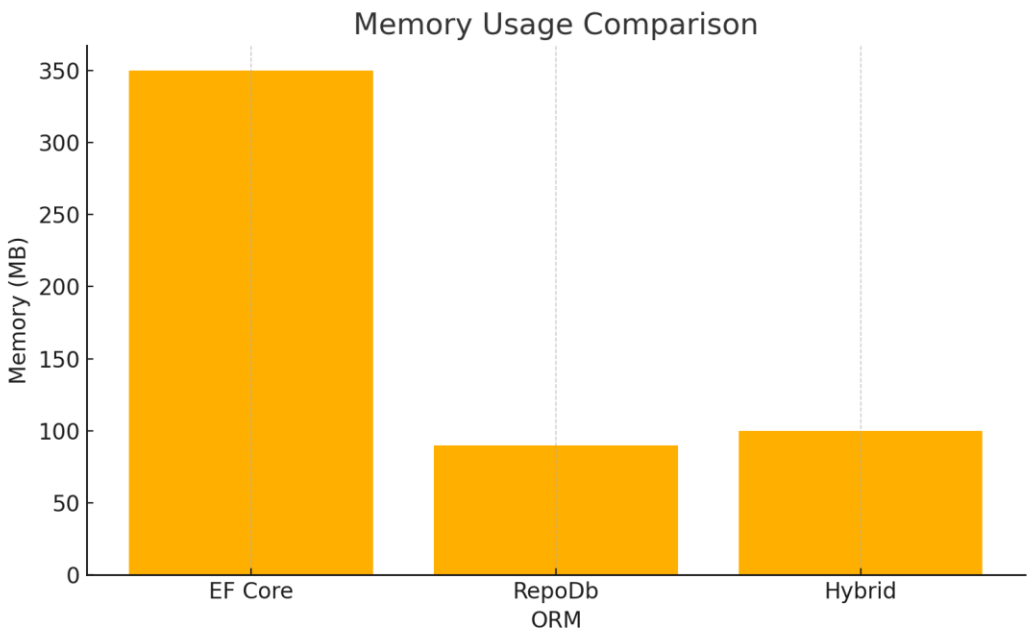

Chart: Memory Usage Comparison (MB)

Evaluation of Other ORMs and Final Insight

ORM	Key Advantages	Limitations or Drawbacks
EF Core	Powerful, Full LINQ and Migration Support, Change Tracking	Slower CRUD, High Memory Overhead
RepoDb	Extremely Fast, Lightweight, No Overhead	No Tracking, No IQueryable Support
Dapper	Lightweight and Fast, Minimal Dependencies	No complex model support, No Migration
NHibernate	Highly Flexible, Advanced Features (Caching, Strategy)	Complex Setup, Lower Performance
Linq2Db	Efficient, Lightweight, Supports Raw SQL	Smaller Community, Limited in Features

Conclusion

Based on the benchmarking and architectural analysis, the hybrid combination of EF Core and RepoDb proves to be highly effective for real-world .NET applications. This approach is use-case driven and optimally assigns tasks:

- EF Core is applied for complex logic, multi-layered relationships (via Include), and schema migrations.
- RepoDb is employed for high-performance CRUD operations and stateless microservices.

As a result, the Hybrid ORM Architecture delivers superior performance, maintainability, and architectural scalability, making it a pragmatic and professional strategy for enterprise-grade software solutions in .NET.