

Технологический стек

1. Backend

1.1 Backend

Технологии

Технология	Версия	Статус	Обоснование
C#	-	Поддерживать	Версия языка определяется используемой версией .NET
.NET Framework	4.8	Поддерживать	Финальная версия платформы с поддержкой до 2028 года

Обоснование: для легаси продуктов приоритетом является стабильность и минимизация рисков изменений. .NET Framework 4.8 является финальной версией платформы и будет поддерживаться Microsoft до 2028 года.

Базы данных

Технология	Статус	Обоснование
MS SQL Server	Мигрировать постепенно	Используется в легаси, при плановом рефакторинге рекомендуется переход на PostgreSQL
PostgreSQL	Поддерживать	Универсальная production-ready СУБД с активным развитием и широкой поддержкой
SQL Compact	Мигрировать	Устаревшая БД, ограниченная функциональность и сильная зависимость от Windows
SQLite	Поддерживать	Легковесная встраиваемая БД для локального хранения данных

Обоснование: PostgreSQL рекомендуется как целевая платформа для новых и модернизируемых компонентов. SQL Compact признана устаревшей и требует поэтапной

миграции без полного переписывания системы.

ORM ? ?????? ? ??

Библиотека	Версия	Статус	Обоснование
Entity Framework	-	❑ Поддерживать	Используется в проектах на .NET Framework, версия определяется поддерживаемой платформой
ADO.NET	-	⚠ Мигрировать постепенно	Прямая работа с БД в легаси системах без изменения архитектуры

Обоснование: для легаси-систем допускается постепенный отказ от прямого использования ADO.NET в пользу Entity Framework. ADO.NET остается допустимым для сложных SQL-сценариев, bulk-операций и модулей без планового рефакторинга. Полный отказ от ADO.NET не является целью; приоритетом является повышение поддерживаемости кода без увеличения рисков и с сохранением возможности повторного использования наработок при дальнейшей модернизации системы.

????????????

Библиотека	Статус	Обоснование
Serilog	❑ Поддерживать	Современная библиотека, поддерживает структурированные логи, гибкие sinks, интеграцию с облачными и аналитическими системами (ELK, Seq, Splunk)
NLog	⚠ Мигрировать постепенно	Рабочее решение для легаси, поддерживает различные targets, но уступает Serilog по структурированности и интеграции с современными системами
log4net	❑ Мигрировать	Устаревшая библиотека, сложно работать с разными типами вывода, не поддерживает современные аналитические стеки (например ELK), рекомендуется мигрировать на Serilog

Обоснование: Для новых модулей и постепенной миграции рекомендуется Serilog — она обеспечивает структурированные логи, гибкие sinks, поддержку облачных и аналитических систем и упрощает централизованное логирование. Это повышает поддерживаемость и интеграцию кода, при этом не требует срочной полной переписки существующего логирования.

????????????

Библиотека	Статус	Обоснование
Newtonsoft.Json	Поддерживать	Замена не критична
Google.Protobuf	Поддерживать	Эффективная бинарная сериализация

Обоснование: Для легаси-систем Newtonsoft.Json остаётся стандартом сериализации. Замена на System.Text.Json не требуется, если нет критичных требований к производительности.

????????? ? ??????????????

Библиотека	Статус	Обоснование
M2Mqtt	Рассмотреть обновление	Устаревающая библиотека MQTT, рекомендуется плановая замена на MQTTnet при модернизации системы
SSH.NET	Поддерживать	Стабильная библиотека для работы по SSH, безопасно использовать в легаси
FluentFTP	Поддерживать	Активно поддерживается, надежная реализация FTP
EGTS	Поддерживать	Специфичный протокол для ГЛОНАСС, поддержка критична для существующих модулей

Обоснование: M2Mqtt устаревает и имеет ограниченное развитие, поэтому для новых модулей или при рефакторинге рекомендуется использовать MQTTnet. Для существующих легаси систем текущая библиотека работает стабильно, срочной замены не требуется. Остальные протоколы остаются актуальными и безопасными для использования.

2. ????? ???? ?????? ??????????????

2.1 Backend

????? ? ????????????

Технология	Версия	Обоснование
.NET	10 (LTS)	LTS-версия с долгосрочной поддержкой (до 14.10.2028), оптимальна для новых production-проектов

Технология	Версия	Обоснование
C#	Актуальная (в составе .NET 10)	Синхронизирована с платформой, новые конструкции упрощают код
ASP.NET Core	10.x	Высокопроизводительный веб-фреймворк, стандарт для backend на .NET

Обоснование: .NET 10 и C# 14 обеспечивают улучшенную производительность, кроссплатформенность, новые языковые конструкции, расширенные стандартные библиотеки и интеграцию с облаком и микросервисами. Долгосрочная поддержка снижает риски техдолга.

???? ??????

Технология	Обоснование
PostgreSQL	Универсальное решение
SQLite	Лёгкая встроенная БД для локальных решений

ORM ? ?????? ? ??

Библиотека	Версия	Обоснование
Entity Framework Core	Актуальная версия для используемой .NET платформы	Поддерживает последние возможности ORM, высокую производительность и совместимость с актуальными инструментами

?????????????

Библиотека	Версия	Обоснование
Serilog	Актуальная версия для .NET 10	Современное структурированное логирование, лёгкая интеграция с облачными и аналитическими системами (ELK, Seq, Splunk), поддержка централизованного логирования

Обоснование: Serilog — стандарт для новых .NET проектов. Поддерживает структурированные логи, гибкие sinks и интеграцию с системами мониторинга, повышая удобство эксплуатации и поддерживаемость кода.

?????????????

Библиотека	Обоснование
System.Text.Json	Встроенная в .NET, высокая производительность

Библиотека	Обоснование
MemoryPack	Для высокопроизводительной бинарной сериализации, оптимально при рефакторинге корпоративного протокола RITM
Google.Protobuf	Эффективная бинарная сериализация, только по согласованию

Обоснование: Для новых проектов рекомендуется использовать System.Text.Json как стандартную библиотеку сериализации: она встроена в .NET, быстрее Newtonsoft.Json, поддерживает современные конструкции Span/Memory и структурированные JSON-схемы. MemoryPack подходит для высокопроизводительных бинарных сценариев, в том числе при переходе с RITM.

????????? ? ??????????????

Библиотека	Обоснование
MQTTnet	Современный MQTT клиент (вместо M2Mqtt)
SignalR	Real-time коммуникация
NModbus	Протокол Modbus)

????????????????? ?????????? ? ??????????????????

CQRS / Mediator

Библиотека	Обоснование
MediatR	Реализация паттерна Mediator для разделения команд и запросов
□ TODO	Рассмотреть переход в сторону бесплатного и поддерживаемого решения

??????????

Библиотека	Обоснование
FluentValidation	Fluent API для проверки сложных условий, повышает читаемость и поддерживаемость кода
□ TODO	Рассмотреть библиотеки для реализации Result паттерна

????????? / DTO

Библиотека	Обоснование
------------	-------------

AutoMapper	Автоматический маппинг объектов, уменьшает ручной код при трансформации DTO и моделей
Mapster	Быстрая и лёгкая альтернатива AutoMapper с source generators и меньшей конфигурацией

“ Mapster, подходит ли для проектов, протестировать source generators и сложные сценарии маппинга

???????????? / ?????????

Библиотека	Обоснование
Polly	Реализация retry, circuit breaker и других политик устойчивости

DI / ????????? ?????????????

Библиотека	Обоснование
Microsoft.Extensions.DependencyInjection	Стандартный контейнер DI для .NET
Scrutor	Расширение для автоматической регистрации сервисов по соглашению, удобно для модульных проектов

??????? ? ???????? ??????

Библиотека / Инструмент	Обоснование
Microsoft.Extensions.Hosting	Управление жизненным циклом приложений, интеграция с DI и background-сервисами
IHostedService / BackgroundService	Простые фоновые задачи внутри приложения, периодичность через Timer/Task.Delay
Quartz.Extensions.Hosting	Планировщик с гибким расписанием и поддержкой критичных задач

- “ Обоснование использования:
- IHostedService / BackgroundService** — для простых фоновых задач, связанных с приложением.
 - Quartz** — для сложного расписания, устойчивости и управления job-ами.

????????????????????????????????

Библиотека	Обоснование
GMap.NET.Core	Работа с картами и геоданными
GeoJSON.Net	Работа с форматом GeoJSON
DotSpatial.Projections	Работа с пространственными проекциями
NetTopologySuite	Работа с геометрией и пространственными данными

???????????

Библиотека	Обоснование
xUnit	Фреймворк для unit-тестов
Moq	Создание моков для зависимостей
AutoFixture.Xunit	Генерация тестовых данных
MockQueryable.Moq	Моки для IQueryable

???????????

- Для **легаси систем** главное — стабильность, постепенная миграция и сохранение существующих наработок.
- Для **новых проектов** — использовать современные, поддерживаемые технологии.