|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Вопрос/Тема** | **Решение** | **Дополнительная инфа** |
| 1 | Технология | 1) .NET 8.0  2) .NET 6.0 | Тип приложения – REST Web API |
| 2 | База данных | СУБД - Oracle, PostgreSQL  1) Entity Framework Core ORM высокого уровня 2) Dapper ORM низкого урвоня  3) ADO.NET | 1) Абстракция, маппинг объектов, миграция, разработка на основе кода  2) Маппинг объектов, быстрота, больший контроль над взаимодействием с базой данных  3) Отсутствие абстракции, маппинга объектов, полный контроль, быстрота, поддержка Oracle специфичных ref\_cur |
| 4 | Конфигурации | 1) Локальные конфигурации  2) Consul при первичном запуске  3) Обращения в Consul при каждом использовании использовании конфигурации 4) Как в 3 варианте, но с кэшированием в озу на определенную длительность | 1) Быстро, не динамично, не безопасно  2) Быстро, не динамично  3) Медленне из за постоянных запросов при каждом использовании, риск недоступности Consul  4) Быстрее 3-го за счет кэишрования, но не нулевой риск с недоступностью Consul |
| 5 | Логирование | Serilog глобальное логирование  1) Логирование прямо в консоль или в файл  2) Логирование прямо в Elastic через запросы в FluentD | 1) Быстрая запись – медленная доставка, низкая нагрузка на само приложение, риск потери логов из за большого количества логов  2) Быстрая запись – быстрая доставка, более высокая нагрузка, вероятность потери логов из за сетевых проблем |
| 6 | Авторизация | 1) Сетевой доступ  2) ApiKey  3) Basic  4) Bearer jwt + asp.net core identity  5) Oauth OpenID + Keycloak | 1) Для внутренних сервисов  2) Для внутренних сервисов, с мониторингом активности обращении  3) Для внутренних и внешних сервисов, с авторизацией на основе учетных данных по заголовку  4) Авторизация на основе токена, с внешним авторизационным ресурсом или со своим ресурсом на каждом сервисе. Также поддержка рефреш токенов для снижения количества обращения по авторизации  5) Oauth + OpenID авторизация на основе токенов с внешним ресурсом KeyCloak, для внешних сервисов |
| 7 | Тестирование | 1) xUnit + Moq  2) xUnit + Moq + TestServer | 1) Модульные тесты  2) Интеграционные тесты |
| 8 | Мониторинг | 1) Метрики и трассировки OpenTelemetry прямо в Elastic rest  2) HealthCheck | 1) Начиная с 8 версии, нативная поддержка OpenTelemetry которая предоставляет детализованные данные по обработке запросов и поведению сервиса 2) Ендпоинт который возвращает информацию по работоспособности сервиса и его зависимостей, может быть настроен постоянно возвращать 200 код ответа |
| 9 | Документация | 1) Swagger  2) Автоматический AI генератор документации | 1) Swagger ui в среде разработки или при локальном дебаге  2) Инструмент который генерирует документацию по API на основе swagger/кода |

|  |  |
| --- | --- |
| **№** | **Дополнительные инструменты и доработки** |
| 1 | 1) Http rest запросы - http clients + Polly библиотека с встроенной реализацией паттерна Circuit breaker  2) Soap xml запросы –добавление WCF зависимости с динамичным url |
| 2 | 1) DTO  2) Mapping – AutoMapper  3) Validation – FluentValidation или атрибуты  4) Repositories – Абстракция работы с базой |
| 3 | Кэширование:  1) Redis  2) Memory cache |
| 4 | Брокер сообщений:  1) Kafka |