МК Музичка архива — Технички преглед

Верзија: 1.0
Статус: Јавен информативен документ
Датум: 11 септември 2026
Категорија: ТЕХНОЛОГИЈА / 04
Авторитет: Информативен. Овој документ ја објаснува техничката основа за заинтересираната јавност и не ги заменува авторитативните технички спецификации на МК Музички регистар.

септември 2026

СОДРЖИНА

1. Зошто технологијата е важна....3
2. Основната архитектура....3
3. Технолошки стек....3
4. Регистарот е авторитативен, јавниот веб е проекција....4
5. Податоци, идентитет и историја....4
6. Безбедност вградена во дизајнот (Security by Design)....4
7. Конфигурација и околини....5
8. Тестирање и контролни точки за квалитет....5
9. Испорака, операции и можност за обновување....6
10. Набљудливост без изложување на чувствителни податоци....7
11. Одржливост и идно скалирање....7
12. Инженерска дисциплина и развој потпомогнат од вештачка интелигенција....7
13. Што овој документ намерно не објавува....8
Управување со документот....8

1. Зошто технологијата е важна

МК Музички регистар е долгорочен систем за културна меморија. Затоа техничката архитектура е дизајнирана пред сè за интегритет на податоците, следливост, безбедност, одржливост, можност за обновување и можност системот да се пренесе во друга средина.

Проектот намерно избегнува непотребна инфраструктурна сложеност. Регистарот започнува како еден добро структуриран систем со јасни внатрешни граници, наместо како збир од повеќе независни сервиси без докажана потреба.

Водечкиот принцип е едноставен: наједноставната архитектура што ги исполнува културните, безбедносните и оперативните барања е подобра од посложена архитектура без измерена потреба.

2. Основната архитектура

Основните дизајнерски и архитектски одлуги се темелат на балансирање помеѓу брза првична испорака и поставување добра основа што ќе овозможат долгогодишна одржливост и развој.

МК Музички регистар се развива како модуларен монолит: една главна апликација и една трансакциска PostgreSQL база, со јасно одделени модули и одговорности.

Основните технички слоеви се:

- **Domain** — деловните и културните концепти, ентитети, објекти на вредности (value objects) и инваријанти (invariants).
- **Application** — експлицитните случаи на употреба (use cases), оркестрација, валидација и договори меѓу модулите.
- **Infrastructure** — PostgreSQL/EF Core persistence, ASP.NET Core Identity, склад (storage), адаптери за интеграција (integration adapters) и друга техничка инфраструктура.
- **Web** — Blazor презентацијата и композицијата на апликацијата.
- **DatabaseMigrator** — контролиран еднократен процес за ажурирање на базата на податоци (production database migrations).

Зависностите се намерно еднонасочни. Domain не зависи од базата на податоци, корисничкиот интерфејс (UI) или веб рамката (web framework). Application не зависи од Web. UI компонентите не пристапуваат директно до слојот за зачувување (persistence).

3. Технолошки стек

Главната технолошка основа е:

- NET 10 и C# 14
- ASP.NET Core
- Blazor Web App со Interactive Server
- Entity Framework Core

- PostgreSQL со Npgsql
- ASP.NET Core Identity
- MudBlazor за back-office презентацијата
- Docker Compose за production topology
- Caddy како HTTPS reverse proxy
- GitHub како извор на код, техничка документација и engineering history

Во MVP (Minimum Viable Product) намерно не се воведуваат микросервиси (microservices), Kubernetes, дистрибуиран посредник за пораки (distributed message broker), Elasticsearch или посебен механизам за работниот тек (workflow engine). PostgreSQL е централна трансакциска платформа и, додека мерливите потреби не покажат поинаку, неговите search, trigram, indexing и transaction можности се доволни за регистарот.

4. Регистарот е авторитативен, јавниот веб е проекција

Регистарот е изворот на структурираното, одобрено знаење.

Јавното претставување не чита директно од интерните записи на регистарот. Системот создава јавни проекции (Public Projections) - избрани, јавни и безбедни претставувања на одобрено знаење во регистарот. Одобрувањето и објавувањето се различни чекори.

Планираната WordPress интеграција е посебен јавен презентациски систем (public presentation system). Таа може да прима и прикажува различни верзии од јавните проекции (versioned Public Projections), но не може да авторизира факти во регистарот, да одобрува записи или да ја менува интерната истражувачка состојба - WordPress базираниот веб сајт служи исклучиво за прикажување на дел од податоците во регистарот, претходно одобрени за јавно презентирање.

5. Податоци, идентитет и историја

PostgreSQL ја чува авторитативната структурирана состојба. Регистарот користи стабилни интерни идентитети за истиот ентитет да остане следлив низ исправки, нови извори и идни интеграции.

Историјата не е само технички додаток за ревизија (audit). Тековната состојба што може да се менува, ревизиите, одобрените слики со податоци (Approved Snapshots), одлуките и настаните за ревизија треба да овозможат да се објасни не само што е прифатено, туку и како се стигнало до тоа знаење.

Датотеките (прилозите) и доказниот материјал не се третираат како обични ресурси директно достапни од веб нивото. Пристапот оди преку апстракција на складот (storage abstraction) и правила за права на пристап и видливост.

6. Безбедност вградена во дизајнот (Security by Design)

Безбедносниот модел е намерно конвенционален и проверлив.

Клучни принципи се:

- недозволен пристап по дефиниција (deny by default);
- минимум привилегии (least privilege);
- авторизација на серверска страна (server-side authorization);
- одделување на човечки, машински, доменски и историски идентитети;
- приватни и ограничени докази не стануваат јавни само затоа што постои јавна проекција (Public Projection);
- постојна и востановен механизам како рамка (framework) се претпочитаат пред сопствено развиена криптографија (custom cryptography);
- тајни и чувствителни податоци не смеат да се појавуваат во код, обични логови или јавни дијагностики.

Бej офис алатката (Back office) користи ASP.NET Core Identity, авторизација базирана на правила (policy-based authorization) и задолжителна TOTP MFA (Time-based One-Time Password Multi-Factor Authentication) за привилегирани корисници според одобрената безбедносна спецификација.

Во продукциската околина само потребните веб порти се јавно изложени преку reverse proxy. PostgreSQL, интерет порт на регистарот (Registry internal port), заштитен склад (protected storage) и оперативни интерфејси (operational interfaces) не се директно изложени на интернет.

7. Конфигурација и околини

Одобрени се точно три апликациски околини:

- Развојна (Development)
- Тест (Testing)
- Продукциска (Production)

Постојана Стејџинг околина (Staging environment) не е дел од MVP.

Конфигурацијата користи framework-native .NET механизми, typed Options и startup validation. Тајните податоци не се пренесуваат во Git (контролата за верзионирање), не се пренесуваат преку командната линија (command line) и не смеат да се појават во тестови, логови, слики од екранот (screenshots) или пријави (tickets). Development користи .NET User Secrets, а Production претпочита key-per-file и надворешно снабдување со secrets.

Тестирањето мора да остане целосно одвоено од продукциските ресурси.

8. Тестирање и контролни точки за квалитет

Квалитетот не се докажува само со успешно компајлирање на изворниот код (build).
Тестирањето е слоевито:

- Domain tests — чисти доменски правила и invariants.

- Application tests — случаи на употреба (use cases), оркестрација (orchestration), авторизација и валидација (authorization and validation).
- Integration tests — реален PostgreSQL преку Testcontainers за зачувување (persistence), миграции (migrations), конкурентност при извршување (concurrency) и специфично однесување на базата на податоци (database-specific behavior).
- Architecture tests — dependency и layer граници.
- Web tests — bUnit за компоненти и Playwright за критични end-to-end патеки.
- Contract, security, accessibility и оперативни проверки — каде што се релевантни.

Проектот користи таканаречен „Fast Gate“ тестови по значајни промени во решението и „Full Gate“ скрипти за тестирање пред поголема функционалност да се смета за завршена. Подготвеност за продукција (Production readiness) бара и оперативни докази (operational evidence), како backup/restore и health verification.

SQLite и EF Core InMemory намерно не се користат како замена за семантиката на PostgreSQL.

9. Испорака, операции и можност за обновување

Продуциската околина е дизајнирана за да не биде зависна од хост провајдерот (provider-neutral Linux/VPS deployment).

Високото ниво на топологија е:

Internet → HTTPS reverse proxy → Registry application → PostgreSQL / protected storage

Регистарот комуницира и со посебниот јавно достапен систем за објавување информации (WordPress public system) преку контролирана граница на HTTPS-интеграција.

Продукциската околина користи Docker Compose, Caddy, контејнер за регистарот, PostgreSQL, заштитено трајно складирање (protected persistent storage), постојани клучеви за заштита на податоци (persistent Data Protection keys) и надворешна поставка за конфигурација (configuration/secrets).

Кратки и контролирани периоди за одржување се прифатливи за MVP. Автоматското префрлување при откажување (Automatic failover) и испорака на нова верзија без прекин во работата (zero-downtime deployment) не се опфат на абота во оваа фаза.

Практичната цел за MVP подразбира RPO (Максимална старост на датотеките што мора да се вратат од резервната меморија (backup) за да продолжат нормалните операции доколку дојде до прекин во работата на компјутер, систем или мрежа.) до 24 часа и RTO (Recovery Time Objective - Максималното прифатливо времетраење во текот на кое деловниот процес или системот може да биде вон функција по дефект или катастрофа.) во рок од еден работен ден по дефект на поединечен VPS (Virtual Private Server) што дозволува обновување. Станува збор за оперативни цели, а не за гаранција за висока достапност.
Резервна копија (Backup) не се смета за докажан додека успешно не е вратен (restore).

10. Набљудливост без изложување на чувствителни податоци

Регистарот користи структурирани оперативни логови, проверки на здравјето/подготвеноста (health/readiness) и контролирана оперативна дијагностика.

Логовите треба да содржат безбеден технички контекст како што се корелациски ID-а, безбедни регистарски ID-а, верзии на проекции, статус, времетраење и состојба на објавување/работа. Тие не смеат да содржат лозинки, MFA тајни, заглавија за авторизација (Authorization headers), колачиња (cookies), приватни податоци на соработници, ограничени докази, содржина на заштитени датотеки или други тајни.

Мониторингот се фокусира на материјални оперативни ризици: достапност на регистарот, PostgreSQL, капацитет на дискот, здравјето на резервните копии (backup), заглавена работа во позадина, неуспеси во објавувањето и релевантни безбедносни услови.

11. Одржливост и идно скалирање

Архитектурата е оптимизирана за мал тим и реален почетен обем, но не е еднократен прототип. Патот за раст е постепен:

1. зголемување на VPS ресурси;
2. издвојување на PostgreSQL;
3. преместување на датотеки во објектно складирање, NAS или институционално складирање;
4. одвојување на заднинските процеси кога измереното оптоварување го оправдува тоа;
5. додавање повеќе инстанци на апликацијата на регистарот само кога тоа е оправдано со реални потреби за истовремено извршување или достапност.

Истиот систем може да се премести на институционална Linux-инфраструктура или на инфраструктура во сопствените простории (on-premises) без потреба од редизајнирање на доменот. Нова архитектонска сложеност се воведува само кога за тоа ќе се покаже потреба.

12. Инженерска дисциплина и развој потпомогнат од вештачка интелигенција

Кодот и техничката документација се организирани во еден GitHub монорепозиториум.

Проектот по дифолт претпочита експлицитни случаи на употреба и локално мапирање, наместо дополнителни рамки за апстракција. MediatR, AutoMapper, генеричкі репозиториум, генеричка единица за работа, BaseService и BaseRepository намерно не се одобрени како стандардни шеми.

Ова го прави системот полесен за разбирање и за луѓето и за агентите за кодирање со вештачка интелигенција.

Вештачката интелигенција може да помогне во имплементацијата, тестирањето и документацијата, но работи во рамките на одобрени спецификации од човек, правила за репозиториум и контролни точки за квалитет. Агентите базирани на вештачка интелигенција, не смеат независно да измислуваат нови архитектонски слоеви, безбедносни модели или семантика на домени.

13. Што овој документ намерно не објавува

Овој Технички преглед не е инженерски прирачник и не ги заменува техничките спецификации.

Намерно ги исклучува:

- шемите на базата на податоци, колони и индекси;
- детални мапирања на ентитети;
- каталози на крајни точки на API или шеми на товар;
- тајни за автентикација или процедури за ракување со акредитиви;
- оперативно детална безбедносна топологија;
- команди за производство, точни патеки или акредитиви за инфраструктура;
- процедури за закрепнување од катастрофа чекор-по-чекор;
- детално дрво за имплементација на складиштето;
- внатрешни политики за авторизација;
- внатрешни тест-инсталации и примери за заштитени податоци.

Негова цел е да покаже дека МК Музички регистар има сериозна, одржлива инженерска основа без да открива детали што не припаѓаат на јавна публикација.

Управување со документот

Овој јавен документ е изведен од одобрените спецификации за: техничка архитектура, безбедност, оперативни процедури (вклучувајќи распоредување и закрепнување од катастрофи), структура на решението и складиштето, стандарди за кодирање, стратегија за тестирање, како и управување со средини, конфигурации и тајни податоци.

Карактер на документот: информативен. Во случај на неусогласеност, одобрените нумерирани спецификации се меродавниот извор на информации за архитектурата, безбедноста, работењето и имплементацијата.

Причини за ажурирање: суштинска промена во архитектурата на апликацијата, основниот технолошки стек, границата за јавна испорака, безбедносниот модел, моделот на средини, контролните точки за тестирање/квалитет, топологијата на продукцискиот систем или целите за закрепнување.