The 10-Point MDR & FHIR Readiness Checklist (2025 Edition)

For HealthTech CTOs & Architects *By Gjergj Sheldija, Principal Healthcare Architect*

Introduction

Most digital health startups fail not because their AI is bad, but because their architecture cannot survive a regulatory audit or a hospital integration request.

If you are building Software as a Medical Device (SaMD) or planning to integrate with German hospital infrastructure (KHZG), use this checklist to self-assess your architectural maturity.

Scoring: If you answer "No" to more than **2 questions**, your product is at high risk of failing an MDR audit or being rejected by hospital IT security.

---

1. The FHIR R4 Standard Check

Is your internal data model strictly mapped to FHIR R4 resources?

- **The Trap:** Storing patient data in custom JSON blobs or proprietary schemas.
- **The Requirement:** You must demonstrate how your internal `Patient`, `Observation`, and `Encounter` objects map to FHIR R4 resources. If you are building for the German market, do you support the **ISiK** (Informationstechnische Systeme in Krankenhäusern) profile?
- **Status:** [ ] Yes / [ ] No

2. Immutable Audit Trails (GDPR & MDR)

Does every single read/write action trigger an immutable log event?

- **The Trap:** Logging only "errors" or "system events."
- **The Requirement:** Every access to Personal Health Information (PHI) must be logged with *Who, What, When, and Why*. These logs must be tamper-proof (write-once) to satisfy GDPR Article 30 and MDR traceability requirements.
- **Status:** [ ] Yes / [ ] No

3. IEC 62304 Documentation

Is your code architecture linked to your Software Safety Classification?

- **The Trap:** treating all code as "Class A" (low risk) to avoid documentation.
- **The Requirement:** If your software influences diagnosis or therapy, it is likely Class B or C. You must segregate high-risk components (e.g., the AI algorithm) from low-risk components (e.g., the UI) to minimize the validation burden.
- **Status:** [ ] Yes / [ ] No

4. Smart on FHIR / OIDC Security

Do you use standard identity providers for authentication?

- **The Trap:** Rolling your own auth or passing credentials in headers.
- **The Requirement:** Hospital IT will insist on **Smart on FHIR** flows (OAuth2/OpenID Connect). Your system must handle token-based context launching (e.g., opening your app *inside* the doctor's EMR with the patient context already loaded).
- **Status:** [ ] Yes / [ ] No

5. The "HL7 v2 Reality" Check

Do you have an adapter layer for legacy inputs?

- **The Trap:** Assuming every hospital uses FHIR.
- **The Requirement:** 90% of German hospitals still run on **HL7 v2 (ADT/ORM/ORU)**. Your architecture must have an integration service (like Mirth/NextGen or a custom Go microservice) that parses legacy pipe-delimited messages and converts them to your internal FHIR model.
- **Status:** [ ] Yes / [ ] No

6. Data Residency & Sovereignty

Can you guarantee data stays in the DACH region?

- **The Trap:** Using US-east-1 AWS buckets or cloud services that implicitly replicate data across borders.
- **The Requirement:** For DiGA (Digital Health Applications) and KHZG funding, strict data residency within the EU (often specifically Germany/C5 criteria) is mandatory.
- **Status:** [ ] Yes / [ ] No

7. Encryption "In Transit" is Not Enough

Do you practice field-level encryption or strict encryption at rest?

- **The Trap:** Relying solely on HTTPS/TLS.
- **The Requirement:** Databases must be encrypted at rest. For sensitive fields (HIV status, mental health), best practice suggests application-level encryption where the database admin cannot read the raw data.
- **Status:** [ ] Yes / [ ] No

8. Requirements Traceability (The V-Model)

Can you trace a line from Code -> Test -> Requirement?

- **The Trap:** Having code that doesn't belong to a documented requirement.
- **The Requirement:** In an audit, "Unreferenced Code" is dead code. You must show that *Feature X* satisfies *Requirement Y* and passed *Test Z*. Automated tooling (e.g., Xray for Jira) helps, but the architecture must support it.
- **Status:** [ ] Yes / [ ] No

9. Disaster Recovery (RTO/RPO)

Have you tested your restore process in the last 6 months?

- **The Trap:** Having backups but never testing the restore time.
- **The Requirement:** Hospitals require defined Recovery Time Objectives (RTO). If your system goes down, can you restore the exact state of patient data from 15 minutes ago?
- **Status:** [ ] Yes / [ ] No

10. API Versioning Strategy

Is your API versioning explicit?

- **The Trap:** Breaking changes in a live clinical environment.
- **The Requirement:** Once you integrate with a hospital, you cannot just "deploy updates." You need a strict versioning strategy (Semantic Versioning) where the hospital can stay on `v1.2` while you release `v2.0` to others.
- **Status:** [ ] Yes / [ ] No

The Verdict

- **0-2 "No" answers:** You are in good shape.
- **3-5 "No" answers:** Your architecture has technical debt that will delay your MDR certification by 3-6 months.
- **6+ "No" answers:** You are at critical risk. Do not submit for audit.

Need a Second Pair of Eyes?

I specialize in fixing the "No" answers.

Gjergj Sheldija *Principal Healthcare Architect* Book a 2-Day Red Flag Audit Email Me