AUDIT-11 — Befunde

Stand: 2026-05-18 (Initial-Lauf)

Offene Befunde

ID	Klausel	Schwere	Befund	Mitigation	Ter
AUDIT-11-F-02	OWASP ASVS V5.1.5	MED	Path-Traversal mit Sonderzeichen-codiertem case_id (V3) — Risiko geringer als V1, aber theoretisch ausnutzbar	case_id-Validator (regex `^[a-z0-9_-]{1,80}$`); FastAPI-Path-Param-Validator	202
AUDIT-11-F-03	DSGVO Art. 32 + ISO 27001 A.12.4	MED	Pipeline-Logs enthalten unanonymisierte PII (V9) — auch AUD-1 / AUDIT-08-F-01	gekoppelt mit AUD-1 (Anonymisierungs-Filter im Logging-Pfad)	202 (mit AUD
AUDIT-11-F-04	Konzeptionell	LOW	Session-Token hat keinen separaten berater_id-Claim — funktioniert via username, bricht bei Multi-User-pro-Koordinator (V10)	Token-Schema erweitern: `username:berater_id:rolle:ts:hmac`; Migration mit Token-Rotation	näch Audi Zykl 28)

Zusammenfassung

Schwere	Offen	Geschlossen	Total
HIGH	0	1	1
MED	2	0	2
LOW	1	0	1

Geschlossene Befunde (Audit-Trail, append-only)

Konvention: Jeder geschlossene Befund hinterlässt mindestens: - Datum der Schließung - Verifikations-Form (Code-Review / Pytest / Re-Audit-Lauf) - Commit-Hash + Datei-Pfad des Fix - Bewertung Restrisiko

So bleibt die Spur „Audit-Lücke gefunden → Sprint gefixt → Re-Audit verifiziert" auch nach Jahren lesbar.

Format-Vorlage (für künftige Schließungen)

```
 | ID | Geschlossen am | Verifikation | Fix-Commit | Restrisiko |
|---|---|---|---|---|
| AUDIT-11-F-XX | YYYY-MM-DD | [Re-Audit-Lauf YYYY-MM-DD / Pytest test_x.py / Code-Review] |
```

Aktueller Stand 2026-05-18 — Sofort-Fix in Folge des Initial-Audits

Der Initial-Self-Audit am 2026-05-18 hat den HIGH-Befund AUDIT-11-F-01 gefunden — die Code-Inspektion zeigte, dass die Vorgangs-Endpoints im Portal-Backend keinen Koordinator-Owner-Check enthielten. Der Fix wurde am **selben Tag** gezogen, statt auf den planmäßigen Termin 2026-05-25 zu warten, weil der Aufwand niedrig (~2 h) und das Risiko hoch war.

ID	Geschlossen am	Verifikation	Fix-Pfad	Restrisiko
AUDIT-11-F-01	2026-05-18	Pytest `tests/test_portal_tenant_isolation.py` mit 9 Tests grün (Operator/Koordinator/Hersteller × Owner-Match/Cross-Tenant)	Commit `751ece85` — `ce-doc-portal/backend/api.py` neue Funktion `_assert_tenant`; `_lade_state` nimmt optionales `request`; Tenant-Filter in `vorgaenge_liste`; 13 Endpoint-Stellen ergänzt	keines — Code-Pfad geschlossen, Test-Suite hängt am CI-Gate

Story für externe Auditoren / Koordinator: 1. Audit am 2026-05-18 gestartet als Soll-Ist-Vergleich zur Behauptung „tenant-isoliert" auf der Vertraulichkeits-Seite 2. Code-Inspektion fand Lücke: `/api/vorgang/{case_id}` ohne Owner-Check 3. Fix gleichentags: `_assert_tenant`-Helper + Integration in `_lade_state` + Listen-Filter 4. Pytest-Suite mit 9 Vektoren verifiziert (1 V1, 1 V4/V5-Klasse, plus Edge-Cases Hersteller/Rolle/Email-Case) 5. Audit-Eintrag append-only — die Lücke und der Fix bleiben dauerhaft dokumentiert

Das ist genau die Geschichte, die ein CE-Koordinator im Pitch hören möchte: nicht „wir sind sicher", sondern „wir haben gegen uns selbst geprüft, einen Fehler gefunden, ihn gefixt — hier ist die Spur".

Bewertung

Der HIGH-Befund AUDIT-11-F-01 ist **konkret und reproduzierbar**, gleichzeitig **fix-bar in ~2 Stunden** durch eine einzige Helper-Funktion. Der Bug ist existent, weil die Tenant-Trennung beim P-1-Sprint (Mandanten-DB-Schema) auf der Mandanten-Schicht eingeführt wurde, die Vorgangs-Schicht aber nicht nachgezogen wurde.

Konsequenz für Pilot-Beginn: AUDIT-11-F-01 ist BLOCKER vor erstem produktiven Pilot-Mandanten. Termin **2026-05-25** im Befund-Dashboard mit höchster Priorität.

Die übrigen drei Befunde sind sprintplanbar im Rahmen der normalen Härtungs-Sprints.