**Abschnitt 1**

Quelle: SaaS-Beschreibung.pdf

*Hinweis: Originalinhalte folgen unverändert im Anschluss an diese Abschnittsseite.*



Anforderungsdokument: Lokaltreu SaaS v1.0 Dokumenten-Status: Final Version: 1.4 Datum:

28. September 2025

**1**

**. Einleitung und Geltungsbereich**

Dieses Dokument definiert die fachlichen und nicht-funktionalen Anforderungen für die erste

Version (MVP) des Lokaltreu SaaS. Es dient als verbindliche Grundlage für die technische

Spezifikation, für Konzeption, Entwicklung und Qualitätssicherung. Der geografische

Geltungsbereich ist Deutschland, die Systemsprache ist Deutsch. ( DSGVO-konformes SaaS-

System)

**2**

**. Produktvision und strategische Ziele**

**2**

**.1. Vision**

Lokaltreu ersetzt veraltete Papier-Stempelkarten durch ein maximal vereinfachtes, sicheres

und vollständig digitales Treuesystem, das speziell auf die Bedürfnisse von Inhabern lokaler

Kleinstunternehmen zugeschnitten ist.

**2**

**.2. Zielgruppe**

Das System ist exklusiv für Inhaber von lokalen Dienstleistungsbetrieben konzipiert,

insbesondere Friseursalons, Kosmetikstudios und Nagelstudios. Die Architektur ist auf den

Anwendungsfall des **Einzelunternehmers als alleiniger Administrator** ausgerichtet, für den

eine intuitive Bedienung ohne Einarbeitungszeit entscheidend ist.

**2**

**.3. Design-Prinzipien**

•

**Single-Administrator-Architektur:** Das System ist auf die Bedienung durch eine

einzige, voll verantwortliche Person (den Inhaber) optimiert. Komplexe Rollen- oder

Rechtekonzepte existieren nicht.

•

•

**Radikale Einfachheit:** Jede Funktion muss ohne Anleitung sofort verständlich sein.

Der administrative Aufwand wird auf ein absolutes Minimum reduziert.

**Sicherheit durch Automation:** Da keine manuelle Zwei-Personen-Kontrolle

vorgesehen ist, schützt das System den Inhaber proaktiv durch automatisierte

Warnungen, Benachrichtigungen und unveränderliche Protokolle.

•

**Client-Leichtigkeit (PWA):** Das Kundenerlebnis funktioniert reibungslos über den

mobilen Browser (Progressive Web App), ohne die Hürde einer App-Installation.

**3**

**. Systemarchitektur und Rollenmodell**

**3**

**.1. Akteure**

Das System definiert drei klar abgegrenzte Akteure:

•

•

**Der Inhaber (Rolle: Administrator):** Besitzt die alleinige administrative Kontrolle

über den Mandanten. Greift über ein Web-Dashboard auf alle Funktionen zu.

**Der Mitarbeiter (Rolle: Benutzer):** Besitzt keinen eigenen Account oder Login. Der

Zugriff erfolgt ausschließlich über ein vom Inhaber autorisiertes Gerät. Die

Interaktionsoberfläche ist auf zwei Kernfunktionen beschränkt.

•

**Der Endkunde (Rolle: Gast):** Interagiert anonym und ohne Account über sein

eigenes Smartphone mit dem System.

**4**

**. Funktionale Anforderungen (User Stories)**

**4**

**.1. Inhaber-Funktionen (Admin-Dashboard)**

•

•

•

•

•

**US-1 (Onboarding & Kampagne):** Als Inhaber möchte ich mich in unter 5 Minuten

registrieren und meine erste Treue-Kampagne (z.B. "5 Stempel = 1 Belohnung")

anlegen können, um sofort startklar zu sein.

**US-2 (Geräteautorisierung):** Als Inhaber möchte ich die Registrierung eines neuen

Mitarbeiter-Geräts initiieren können, indem ich einen sicheren, einmalig gültigen Link

generiere und versende.

**US-3 (Geräteverwaltung):** Als Inhaber möchte ich jederzeit ein registriertes Gerät

mit einem einzigen Klick sperren oder löschen können, um bei einem

Mitarbeiterwechsel die volle Kontrolle zu behalten.

**US-4 (Reporting):** Als Inhaber möchte ich ein übersichtliches Dashboard sehen, das

mir die wichtigsten Aktivitäten (vergebene Stempel, eingelöste Prämien) pro Tag,

Woche und Monat anzeigt, um den Erfolg meines Programms zu überblicken.

**US-5 (Angebots-Funktion):** Als Inhaber möchte ich ein optionales, kurzes

Werbeangebot in meinem Dashboard eintragen können, das meinen Kunden nach dem

Stempel-Scan angezeigt wird, um einfach und ohne Zusatzkosten auf Aktionen

aufmerksam zu machen.

•

•

**US-6 (Stempelvergabe):** Als Inhaber möchte ich selbst über mein Dashboard oder ein

autorisiertes Gerät Stempel vergeben können, um Kunden schnell bedienen zu können.

**US-7 (Prämieneinlösung):** Als Inhaber möchte ich über mein Dashboard oder ein

autorisiertes Gerät eine Prämie einlösen können, um den Prozess selbstständig und

sicher abzuschließen.

**4**

**.2. Mitarbeiter-Funktionen (Mitarbeiter-Gerät)**

•

**US-8 (Stempelvergabe):** Als Mitarbeiter möchte ich auf dem autorisierten Gerät mit

einem Klick einen einmalig gültigen QR-Code zur Stempelvergabe erzeugen können,

um den Kunden schnell zu bedienen.

•

**US-9 (Prämieneinlösung):** Als Mitarbeiter möchte ich durch Scannen der digitalen

Stempelkarte eines Kunden eine Prämie als "eingelöst" markieren können, damit der

Prozess eindeutig und sicher ist.

**4**

**.3. Endkunden-Funktionen (PWA)**

•

**US-10 (Stempel sammeln):** Als Endkunde möchte ich durch das Scannen eines QR-

Codes einen digitalen Stempel erhalten, ohne eine App installieren oder persönliche

Daten angeben zu müssen.

•

**US-11 (Prämie vorzeigen):** Als Endkunde möchte ich meine volle digitale

Stempelkarte auf meinem Smartphone vorzeigen können, die einen einmalig gültigen

QR-Code zur Einlösung enthält. (Nur Mitarbeiter oder Inhaber können diesen QR-

Code zur Einlösung scannen und die Prämie einlösen).

•

**US-12 (Angebote sehen):** Als Endkunde möchte ich nach dem Scannen eines

Stempels über aktuelle Angebote des Geschäfts direkt auf meiner digitalen

Stempelkarte informiert werden, um von Aktionen zu profitieren.

**4**

**.4. Kunden-werben-Kunden-System**

US-13: Als Endkunde möchte ich einen personalisierten Link erhalten, um Freunde

einzuladen, und einen Stempel erhalten, sobald der geworbene Freund seinen ersten

Stempel erfolgreich validiert hat.

US-14: Als geworbener Neukunde möchte ich einen Stempel erhalten, sobald ich

meinen ersten Stempel erfolgreich validiert habe.

**5**

**. Detaillierte Prozessabläufe**

**5**

**.1. Prozess: Autorisierung eines neuen Mitarbeiter-Geräts**

Der Prozess wird vom Inhaber initiiert und vom Mitarbeiter auf dem Zielgerät abgeschlossen.

1

. **Initiierung (Inhaber):** Der Inhaber wählt im Admin-Dashboard die Option "Neues

Mitarbeiter-Gerät hinzufügen". Das System generiert einen zeitlich begrenzten (z.B.

1

5 Minuten) und einmalig gültigen Registrierungs-Link. Dieser wird wahlweise als

QR-Code zur direkten Erfassung oder als Text-Link zum Kopieren bereitgestellt.

. **Registrierung (Mitarbeiter):** Der Mitarbeiter öffnet den Link auf dem zu

registrierenden Gerät (via Scan oder Klick). Eine Bestätigungsseite wird angezeigt, die

das Geschäft namentlich nennt.

. **Zustimmung (Mitarbeiter):** Der Mitarbeiter bestätigt die Registrierung des Geräts

für das angezeigte Geschäft durch einen Klick auf einen eindeutigen Bestätigungs-

Button.

. **Abschluss (System):** Das System bindet die eindeutige Gerätekennung kryptografisch

an den Account des Inhabers. Der Registrierungs-Link wird sofort invalidiert. Eine

Erfolgsmeldung wird dem Mitarbeiter angezeigt, und der Inhaber erhält eine

automatische Sicherheits-Benachrichtigung per E-Mail.

2

3

4

**5**

**.2. Prozess: Operative Mitarbeiter-Abläufe**

Der Mitarbeiter greift über ein Lesezeichen (Home-Screen-Icon) auf eine minimalistische

Oberfläche mit zwei Optionen zu:

•

**Ablauf A: Stempel vergeben**

Der Mitarbeiter tippt auf "Neuen Stempel vergeben".

Das System generiert und zeigt einen einmalig gültigen QR-Code

bildschirmfüllend an.

o

o

Der Endkunde scannt den QR-Code mit seinem Smartphone und erhält den

Stempel.

Die Ansicht kehrt nach kurzer Zeit automatisch zum Startbildschirm zurück.

o

o

•

**Ablauf B: Prämie einlösen**

Der Endkunde zeigt seine volle Stempelkarte mit dem "Prämie-einlösen"-QR-

Code vor.

o

Der Mitarbeiter tippt auf "Prämie einlösen", wodurch die Gerätekamera

aktiviert wird.

o

Der Mitarbeiter scannt den QR-Code auf dem Gerät des Kunden.

Das System validiert den Code, entwertet die Kundenkarte und protokolliert

die Transaktion. Eine Erfolgsmeldung wird angezeigt, bevor die Ansicht zum

Startbildschirm zurückkehrt.

o

o



**6**

**. Nicht-funktionale Anforderungen (NFRs)**

**6**

**.1. Sicherheit (MUSS-Anforderungen)**

•

**Warn-Dialog:** Jede kritische, administrative Aktion (z.B. Link-Generierung,

Gerätesperrung) erfordert eine explizite Bestätigung durch den Inhaber in einem

Dialogfeld.

•

•

**Sicherheits-Alerts:** Nach jeder kritischen Aktion wird eine automatisierte E-Mail-

Benachrichtigung an die hinterlegte Adresse des Inhabers versendet.

**Audit-Log:** Alle sicherheitsrelevanten Aktionen (Aktionstyp, Zeitstempel, betroffenes

Gerät) müssen manipulationssicher protokolliert und für den Inhaber im Dashboard

einsehbar sein.

**6**

**.2. Benutzerfreundlichkeit und Performance (MUSS-Anforderungen)**

•

•

Die Admin-Oberfläche muss vollständig selbsterklärend und für mobile Endgeräte

optimiert sein.

Die serverseitige Verarbeitungszeit nach dem Scan eines QR-Codes durch den

Endkunden darf 3 Sekunden nicht überschreiten (p95).

**7**

**. Abgrenzung (Out of Scope für v1.0)**

Die folgenden Funktionen werden in der ersten Version bewusst nicht umgesetzt, um die

Einfachheit zu wahren:

•

Mehrere Administrator-Accounts oder ein komplexes Rollen-Management ("Team-

Modus").

•

•

•

•

•

Ein Vier-Augen-Prinzip für Freigabeprozesse.

Detaillierte Kundenanalysen oder CRM-Funktionen.

Direkter, proaktiver Versand von Mitteilungen (E-Mail, SMS, Push) an Endkunden.

Direkte Integrationen in externe Kassen-Systeme (POS).

Verwaltung mehrerer, parallellaufender Treue-Kampagnen.

**8**

**. Geschäftsmodell und Preispläne**

**8**

**.1. Preispläne**

Die Nutzung von Lokaltreu ist abonnementbasiert und in drei Pläne gestaffelt, die sich an der

Nutzungsintensität und Teamgröße orientieren.

**Plus (Beliebtester**

**Merkmal**

**Preis pro Monat**

**Starter**

**Premium**

**Plan)**

19 €

39 €

69 €

Bis zu **250** pro Bis zu **1.000** pro

Bis zu **2.500** pro

Monat

**Inkludierte Stempel**

Monat

2

Monat

**Mitarbeiter-Geräte**

**Treue-Kampagne**

Bis zu 10

Bis zu 20

✅

✅

✅

✅

✅

**Dashboard & Reporting**

✅

**Funktion: "Aktuelles**

**Angebot"an Kunden senden**

✅

❌

✅

✅

✅

**Kunden-werben-Kunden**

✅

E-Mail-

Support

Priorisierter E-Mail-

Support

**Support**

E-Mail-Support

**8**

**.2. Umgang mit Plan-Überschreitungen**

Um den Geschäftsbetrieb des Inhabers nicht zu unterbrechen, wird die Stempelvergabe bei

Erreichen des monatlichen Limits nicht blockiert. Stattdessen wird ein faires Upgrade-Modell

angewendet:

•

•

**Proaktive Benachrichtigung:** Der Inhaber wird per E-Mail informiert, sobald 80 %

des monatlichen Stempel-Limits erreicht sind.

**Upgrade-Möglichkeit:** Bei Erreichen von 100 % erhält der Inhaber eine weitere

Benachrichtigung mit der Möglichkeit, für den Rest des Abrechnungszeitraums auf

den nächsthöheren Plan upzugraden.

**Abschnitt 2**

Quelle: Technische Spezifikation – Lokaltreu v2.0.pdf

*Hinweis: Originalinhalte folgen unverändert im Anschluss an diese Abschnittsseite.*



Technische Spezifikation – Lokaltreu v2.0

**Version:** 2.0 • **Datum:** 2025-09-29 • **Autor:** Senior Solutions Architect / Technischer

Redakteur

**Änderungsverlauf:**

•

v2.0 – Vollständige, eigenständige Go-Live-Spezifikation. Integriert: hartes Referral-

Gate (Planprüfung UI/Backend), Personenbezug in Betriebs-Logs inkl. Art. 6 Abs. 1

lit. f und Art. 11-Prozess, Scope-Absicherung mit DB-Constraint „eine aktive

Kampagne je Mandant“, Geräte-Onboarding-TTL/Einmaligkeit mit E2E-Protokollen,

Plan-Limits ohne Sperre mit Metriken.

**Inhaltsverzeichnis**

1

2

3

4

5

6

7

8

9

1

1

1

1

1

1

1

1

1

1

2

2

2

2

2

. Deckblatt

. Zusammenfassung & Geltungsbereich

. Ziele, Geschäftsregeln und Design-Prinzipien

. Akteure und Rollenmodell

. Funktionale Anforderungen (User Stories)

. Prozessbeschreibungen & Sequenzen

. Systemarchitektur

. Sicherheitskonzept

. Datenschutz & Compliance (DE/EU)

0. Nicht-funktionale Anforderungen (NFRs)

1. Datenmodell

2. API-Design & OpenAPI 3.1

3. PWA-Umsetzung

4. Geräteverwaltung

5. QR-gestützte Workflows & Anti-Replay

6. Reporting & Dashboard

7. Preispläne & Limits

8. Admin-UI, Mitarbeiter-UI, Endkunden-PWA-UI

9. Betrieb & Deployment

0. Teststrategie, Observability & Nachweise

1. Risiken & Annahmen

2. Rückverfolgbarkeitsmatrix

3. Glossar

4. Anhänge (Artefakte, Runbooks, Berichte)

**1**

**. Zusammenfassung & Geltungsbereich**

**Kurzüberblick.** Digitale Stempelkarte als PWA. Stempel per QR-Scan. Prämien nach

definierter Stempelanzahl. „Kunden-werben-Kunden“ ist rein stempelbasiert: Werber erhält

einen Bonus-Stempel, sobald der geworbene Freund seinen ersten Stempel validiert. Keine

monetären Auszahlungen, keine Zahlungsdienstleister, keine Payout-Integrationen.



**Geltungsbereich.** Deutschland. Systemsprache Deutsch. Verarbeitung in der EU.

**In Scope.** Merchant-Portal, Mitarbeiter-Geräteoberfläche, Endkunden-PWA, QR-Logik,

Gerätebindung, Reporting, Anti-Abuse, Audit, DSGVO, stempelbasiertes Referral (US-

1

3/US-14).

**Out of Scope.** Mehrfach-Admins/Rollen, Vier-Augen-Prinzip, CRM/Marketing-Automation,

proaktive Endkunden-Mailings, POS-Integrationen, **parallele Kampagnen** je Mandant.

**Verifizierbarkeit.** In-Scope/Out-of-Scope decken sich mit UI, API und Datenmodell;

Volltextsuche zeigt keine Payment-Begriffe.

**2**

**. Ziele, Geschäftsregeln und Design-Prinzipien**

•

•

•

•

•

•

**Single-Admin je Mandant (MUSS).**

**Radikale Einfachheit (MUSS):** Mitarbeitergerät hat zwei Kernaktionen.

**Sicherheit durch Automation (MUSS):** Alerts, unveränderliches Audit, Rate-Limits.

**PWA-First (MUSS):** mobil optimiert, installierbar.

**Datensparsamkeit (MUSS):** Endkunden ohne Login; pseudonyme Card-IDs.

**Idempotente Geschäftsaktionen (MUSS).**

**Verifizierbarkeit.** Kein Team-Menü, PWA installierbar, Audit-Events vollständig.

**3**

**. Akteure und Rollenmodell**

•

•

•

**Inhaber (Admin):** Merchant-Portal, volle Konfiguration.

**Mitarbeiter (gerätgebunden, ohne Login):** Stempel vergeben, Prämien einlösen.

**Endkunde (anonym):** Stempel sammeln, Prämie vorzeigen, Referral-Link abrufen.

**Rechte/Angriffsflächen.** Admin-Konto, autorisierte Geräte, anonyme PWA.

**Verifizierbarkeit.** Mitarbeiter-UI hat exakt zwei Aktionen.

**4**

**. Funktionale Anforderungen (User Stories)**

**US-1** Admin registriert Mandant und legt erste Kampagne in < 5 Minuten an. **MUSS**

**US-2** Admin autorisiert Mitarbeitergerät per einmaligem Link (TTL 15 Min). **MUSS**

**US-3** Admin sperrt/löscht Mitarbeitergerät. **MUSS**

**US-4** Admin sieht Aktivitäten (Tag/Woche/Monat). **MUSS**

**US-5** Admin pflegt optionales Angebot nach Scan. **SOLL**

**US-6** Admin/Mitarbeiter initiiert Stempelvergabe. **MUSS**

**US-7** Admin/Mitarbeiter löst Prämie ein. **MUSS**

**US-8** Mitarbeiter erzeugt einmaligen QR-Token. **MUSS**

**US-9** Mitarbeiter validiert Redeem-Token. **MUSS**

**US-10** Endkunde scannt QR und erhält Stempel. **MUSS**

**US-11** Endkunde sieht Karte/Rewards. **MUSS**

**US-12** Endkunde sieht Angebot nach Scan. **SOLL**



**US-13** Werber erhält personalisierten Referral-Link; Werber erhält **+1 Stempel**, sobald der

geworbene Freund seinen **ersten Stempel** erfolgreich validiert. **MUSS**

**US-14** Neukunde erhält seinen **ersten Stempel**; Referral wird „qualifiziert“; **keine**

**Zusatzprämie** für den Neukunden. **MUSS**

**Verifizierbarkeit.** Je US ≥ 2 Gherkin-Szenarien; Trace zu

Prozessen/APIs/Datenobjekten/Tests.

**5**

**. Prozessbeschreibungen & Sequenzen**

**5**

**.1 Onboarding & Kampagne**

sequenceDiagram

participant Admin

participant Portal as Merchant-Portal

participant API

participant DB

Admin->>Portal: Registrieren

Portal->>API: POST /admins/register

API->>DB: Tenant + Default-Kampagne

API-->>Portal: 201 Session

Admin->>Portal: Kampagne konfigurieren

Portal->>API: PUT /campaigns/{id}

API-->>Portal: 200

**5**

**.2 Geräteautorisierung (einmalig, TTL 15 Min, E-Mail-Alert)**

sequenceDiagram

participant Admin

participant Portal

participant API

participant Mail

participant Device

Admin->>Portal: Neues Mitarbeiter-Gerät

Portal->>API: POST /devices/registration-links (TTL=15m)

API-->>Portal: 201 {link, qr}

Device->>API: GET /devices/register?token

API-->>Device: Bestätigungsseite

Device->>API: POST /devices/register/confirm

API->>Mail: Sicherheits-Alert an Admin

**5**

**.3 Ablauf A** – **Stempelvergabe**

sequenceDiagram

participant Staff as Mitarbeiter-UI

participant PWA as Endkunden-PWA

participant API

Staff->>API: POST /stamps/tokens

API-->>Staff: 201 {qrToken,jti,expiresAt}

PWA->>API: POST /stamps/claim {qrToken, ref?}

API->>API: CAS(jti) -> add Stamp (ACID)

API-->>PWA: 200 {cardState, offer?}

**5**

**.4 Ablauf B** – **Prämieneinlösung**



sequenceDiagram

participant PWA as Endkunden-PWA

participant Staff as Mitarbeiter-UI

participant API

PWA->>PWA: Redeem-QR (one-time)

Staff->>API: POST /rewards/redeem {redeemToken}+X-Device-Proof

API-->>Staff: 200 OK

**5**

**.5 Referral** – **stempelbasiert**

sequenceDiagram

participant Werber as PWA (Werber)

participant Freund as PWA (Neukunde)

participant API

Werber->>API: GET /referrals/link

API-->>Werber: 200 {refCodeURL}

Freund->>API: POST /stamps/claim {qrToken, ref=code}

API->>API: if firstStamp(referredCardId) then qualifyReferral +

creditBonusStamp(referrerCardId)

API-->>Freund: 200 {cardState}

**Verifizierbarkeit.** E2E: Normalfall, expired, reuse, tenant-mismatch, self-referral, velocity-

limit.

**6**

**. Systemarchitektur**

**Stil.** Modularer Monolith auf PaaS.

**Technologien.** HTTP API, PWA, **PostgreSQL**, **Redis**, Queue, Mailer, CDN/Static.

flowchart TB

subgraph Client

AdminUI[Merchant-Portal]

StaffUI[Mitarbeiter-UI]

PWA[Endkunden-PWA]

end

CDN[CDN/Static]

API[Monolithische API (PaaS)]

DB[(PostgreSQL)]

REDIS[(Redis)]

QUEUE[(Queue)]

MAIL[Mail-Service]

Client-->CDN

Client-->API

CDN-->API

API-->DB

API-->REDIS

API-->QUEUE-->API

API-->MAIL

**Trust Boundaries.** TLS extern, private Netze intern, **Multi-AZ** für API/DB.

**Verifizierbarkeit.** TLS erzwungen; SG/NACLs dokumentiert.



**7**

**. Sicherheitskonzept**

**7**

**.1 AuthN/AuthZ**

•

**Admin (MUSS):** JWT Access exp ≤ 15 min, Refresh ≤ 30 d, JWKS /.well-

known/jwks.json mit kid.

•

•

**Gerät (MUSS):** Ed25519-Schlüssel; X-Device-Proofüber (method|path|ts|jti).

**Endkunde (MUSS):** anonym, Card-ID pseudonym.

**7**

**7**

**.2 QR/Token**

•

•

•

**Einmaligkeit (MUSS):** jti(UUIDv7).

**TTL (MUSS):** 60 s (+/- 30 s Skew).

**QR-Parameter (MUSS):** ECC „Q“, ≥ 300×300 px, Kontrast ≥ 4.5:1.

**.3 Anti-Abuse**

•

•

**Rate-Limits (MUSS):** Tenant 600 rpm; IP anonym 120 rpm; /stamps/claim 30

rpm/Card; /rewards/redeem 10 rpm/Device.

**Referral-Schutz (MUSS):** Self-Referral blockiert; Velocity-Limit 5 qualifizierte

Referrals/Kunde/Monat; Mandantenbindung.

**7**

**7**

**.4 Fehlertaxonomie & Idempotenz**

•

•

**RFC 7807 (MUSS):** 4

00/401/403/409/422/429/5xx correlation\_id

.

,

**Idempotenz (MUSS):** Idempotency-Key (24 h; Scope

{tenantId,route,bodyHash} ; Retry-Backoff (SOLL).

**.5 Audit**

•

•

**WORM (MUSS):** ts, tenantId, actorType, actorId/deviceId/cardId,

action, target, result, ip, ua, jti, refCode?.

**Events (MUSS):** device.register, stamp.token.issued, stamp.claimed,

reward.redeemed, referral.link.issued, referral.first\_stamp.qualified,

referral.bonus\_stamp.credited.

•

**Aufbewahrung (MUSS):** 180 Tage; signierter Export.

**Verifizierbarkeit.** Replay → 409; Limits → 429/422; Audit vollständig.

**8**

**. Datenschutz & Compliance (DE/EU)**

**8**

**.1 Grundsätze**

•

•

**Datensparsamkeit & Zweckbindung (MUSS).** Endkunden ohne Login.

**Rechtsgrundlage (MUSS).** Art. 6 Abs. 1 lit. f DSGVO für Betrieb, Sicherheit, Fraud-

Prevention.



**8**

**.2 Personenbezug in Betriebs-Logs (NEU, MUSS)**

•

•

**IP-Adressen in Access/Audit-Logs sind personenbezogen.**

**RoPA (MUSS):** Verarbeitungsvorgang „Betriebs-/Sicherheitslogs“ mit Zweck,

Rechtsgrundlage, Löschfristen.

•

•

**TOMs (MUSS):** Rollentrennung, Need-to-know, Protokoll-Schutz, Löschautomatik.

**DSR nach Art. 11 DSGVO (MUSS):** Keine zusätzliche Identifizierung. Vorgehen:

Eingang → Matching auf Card-/Geräte-Kontext ohne Zusatzdaten →

Auskunft/Löschung, soweit möglich → andernfalls Information nach Art. 11(2).

**8**

**.3 Consent-Hinweis (MUSS)**

Endkunden-Interaktionen sind anonym. Kein Einwilligungsbanner erforderlich. Technisch

notwendige Cookies/LocalStorage dienen der Funktionssicherung (Art. 6 Abs. 1 lit. f

DSGVO).

**8**

**.4 Artefakte (MUSS)**

AVV (Art. 28), TOMs, RoPA (Art. 30), DPIA (Art. 35), DE-Rechtstexte (Art. 13/14),

Impressum/Datenschutzhinweise.

**8**

**.5 Lösch-/Aufbewahrungsfristen (MUSS)**

Audit/Alerts 180 Tage; Gerätebindung: Löschung bei Entzug, Protokoll 180 Tage;

Abrechnungsaggregate 10 Jahre (aggregiert), Rohzähler 180 Tage.

**Verifizierbarkeit.** Artefakte versioniert; Löschjobs protokolliert.

**9**

**. Nicht-funktionale Anforderungen (NFRs)**

•

•

•

**Performance (MUSS):** p50 ≤ 500 ms, p95 ≤ 3000 ms, p99 ≤ 6000 ms je Route.

**Verfügbarkeit (MUSS):** SLO 99,90 %/30 d für /

**Resilienz (MUSS):** RPO 15 Min, RTO 60 Min; Multi-AZ; definierte

stamps/claim /rewards/redeem

,

.

Degradationspfade.

•

**A11y-Hinweis:** formale WCAG-Prüfung nicht Bestandteil; Verbesserungen geplant.

**Verifizierbarkeit.** Lasttest-Berichte mit p50/p95/p99; SLO-Dashboard aktiv.

**1**

**0. Datenmodell**

**1**

**0.1 ERD**

classDiagram

class Tenant { id pk; plan; createdAt }

class Device { id pk; tenantId fk; status; boundAt; fingerprint }



class Campaign { id pk; tenantId fk; stampsRequired int; rewardLabel;

active bool }

class StampToken { jti pk; tenantId fk; deviceId fk; expiresAt; usedAt? }

class Stamp { id pk; tenantId fk; cardId; ts; meta jsonb }

class RewardToken { jti pk; tenantId fk; cardId; expiresAt; usedAt? }

class Reward { id pk; tenantId fk; cardId; ts; status }

class Offer { id pk; tenantId fk; title; body; activeFrom; activeTo;

active bool }

class Referral { code pk; tenantId fk; referrerCardId; referredCardId;

firstStampAt?; qualified bool; bonusCreditedAt? }

class AuditLog { id pk; tenantId fk; ts; actorType; actorId; action;

target; result; ip; ua; jti?; refCode? }

class PlanCounter { id pk; tenantId fk; month; stampsUsed; limit;

devicesAllowed }

Tenant <|-- Device

Tenant <|-- Campaign

Tenant <|-- Offer

Tenant <|-- PlanCounter

Tenant <|-- AuditLog

Tenant <|-- StampToken

Tenant <|-- Stamp

Tenant <|-- RewardToken

Tenant <|-- Reward

Tenant <|-- Referral

**1**

**0.2 Schlüssel/Indizes/Constraints (MUSS)**

•

•

•

•

•

•

StampToken.jti UNIQUE (usedAt IS NULL).

UNIQUE (qualified=true) je referredCardId.

Referral

Stamp.meta.reason='referral\_bonus' indexiert.

**Scope-Absicherung (NEU): eine aktive Kampagne je Mandant**:

CREATE UNIQUE INDEX ux\_campaign\_active\_single

ON campaigns(tenant\_id) WHERE active = true;

**Verifizierbarkeit.** Migration erzeugt Indizes/Constraints; negative Tests für

ux\_campaign\_active\_single.

**1**

**1. API-Design & OpenAPI 3.1**

**1**

**1.1 Grundsätze**

REST/JSON, /v1. Fehler application/problem+json (RFC 7807). Idempotenz via

Idempotency-Key (24 h). SecuritySchemes: AdminAuth (JWT), DeviceKey + X-Device-

Proof.

**1**

**1.2 OpenAPI (auszugsweise; vollständig im Anhang)**



`

#

#

` `yaml

Lokaltreu HTTP-API — OpenAPI 3.1 SSOT, v2.0

Kernflüsse, Fehlerformat, Idempotenz, Plan-Gates und Security stammen aus der Technischen Spez

ifikation v2.0 und Roadmap 2.2. :contentReference[oaicite:0]{index=0} :contentReference[oaicite:

]{index=1}

1

openapi: 3.1.0

jsonSchemaDialect: https://json-schema.org/draft/2020-12/schema

info:

title: Lokaltreu HTTP-API

version: "2.0"

description: >

Digitale Stempelkarte als HTTP-API. RFC 7807 für Fehler. Idempotenz via Header.

Referral-Funktionen sind planabhängig (Starter blockiert).

termsOfService: https://lokaltreu.example/terms

contact:

name: Lokaltreu API Support

url: https://lokaltreu.example/support

license:

name: Proprietary

x-schema-drift-policy: "0" # CI-Gate: schema\_drift = 0. :contentReference[oaicite:2]{index=2

}

servers:

url: https://api.lokaltreu.example/v2

description: Produktion EU

tags:

-

-

-

-

-

-

name: Admins

description: Admin-Registrierung

name: Devices

description: Geräte-Onboarding und -Authentisierung

name: Stamps

description: Stempel-Workflows

name: Rewards

description: Prämien-Workflows

name: Referrals

description: Kunden-werben-Kunden

security: [] # Default: öffentlich, Operationen setzen Security explizit

paths:

/

admins/register:

post:

tags: [Admins]

summary: Registriert Administrator und Mandant

description: Erstregistrierung. Liefert Session-Tokens.

security: [] # öffentlich

requestBody:

required: true

content:

application/json:

schema:

type: object

additionalProperties: false

required: [email, password]

properties:

email:

type: string

format: email

password:

type: string

minLength: 12

responses:

"

201":

description: Angelegt

content:

application/json:

schema:

$ref: "#/components/schemas/AdminRegistrationResponse"

"

"

"

"

4 00":

r ef: "#/components/responses/400BadRequest"

4 09":

$

$

ref: "#/components/responses/409Conflict"

4 29":

$

ref: "#/components/responses/429RateLimited"

5 00":

$

ref: "#/components/responses/500ServerError"

#

Sequenz siehe Spezifikation 5.1. :contentReference[oaicite:3]{index=3}

/

devices/registration-links:

post:

tags: [Devices]

summary: Erzeugt einmaligen Registrierungslink (TTL 15 Min)

description: Idempotent. Link ist einmalig und zeitlich begrenzt. :contentReference[oaicit

e:4]{index=4}

security:

-

AdminAuth: []

parameters:

-

$ref: "#/components/parameters/IdempotencyKey"

responses:

"

201":

description: Link erstellt

headers:

Idempotency-Key:

$

ref: "#/components/headers/Idempotency-Key"

content:

application/json:

schema:

$ref: "#/components/schemas/DeviceRegistrationLinkResponse"

"

"

"

"

"

"

4 00":

r ef: "#/components/responses/400BadRequest"

4 01":

$

$

ref: "#/components/responses/401Unauthorized"

4 03":

$

ref: "#/components/responses/403Forbidden"

4 09":

$

ref: "#/components/responses/409Conflict"

4 29":

$

ref: "#/components/responses/429RateLimited"

5 00":

$

ref: "#/components/responses/500ServerError"

/

devices/register/confirm:

post:

tags: [Devices]

summary: Bestätigt Gerätebindung

description: Einmalig verwendbar. Idempotent. Sicherheits-E-Mail wird versendet. :contentR

eference[oaicite:5]{index=5}

security: [] # öffentlich, da Link besitzt Geheimnis

parameters:

-

$ref: "#/components/parameters/IdempotencyKey"

requestBody:

required: true

content:

application/json:

schema:

type: object

additionalProperties: false

required: [token]

properties:

token:

type: string

responses:

"

204":

description: Bestätigt

headers:

Idempotency-Key:

$

ref: "#/components/headers/Idempotency-Key"

"

"

"

"

4 00":

$r ef: "#/components/responses/400BadRequest" # TOKEN\_EXPIRED

4 09":

$

ref: "#/components/responses/409Conflict"

# TOKEN\_REUSE

4 29":

$

ref: "#/components/responses/429RateLimited"

5 00":

$

ref: "#/components/responses/500ServerError"

/

stamps/tokens:

post:

tags: [Stamps]

summary: Erzeugt einmaligen Stempel-QR-Token

description: Idempotent. TTL 60 s. Einmaligkeit per jti. :contentReference[oaicite:6]{inde

x=6}

security:

-

-

DeviceKey: []

AdminAuth: []

parameters:

-

$ref: "#/components/parameters/IdempotencyKey"

responses:

"

201":

description: Token erzeugt

headers:

Idempotency-Key:

$

ref: "#/components/headers/Idempotency-Key"

content:

application/json:

schema:

$ref: "#/components/schemas/StampTokenResponse"

"

"

"

"

4 01":

r ef: "#/components/responses/401Unauthorized"

4 03":

$

$

ref: "#/components/responses/403Forbidden"

4 29":

$

ref: "#/components/responses/429RateLimited"

5 00":

$

ref: "#/components/responses/500ServerError"

/

stamps/claim:

post:

tags: [Stamps]

summary: Anspruch auf Stempel einlösen; qualifiziert ggf. Referral

description: Idempotent. ACID-Claim mit Anti-Replay. Bei ref wird Referral geprüft. :conte

ntReference[oaicite:7]{index=7}

security: []

parameters:

-

$ref: "#/components/parameters/IdempotencyKey"

requestBody:

required: true

content:

application/json:

schema:

$

ref: "#/components/schemas/StampClaimRequest"

responses:

"

200":

description: OK

content:

application/json:

schema:

$ref: "#/components/schemas/StampClaimResponse"

"

"

"

4 00":

r ef: "#/components/responses/400BadRequest" # TOKEN\_EXPIRED

4 09":

$

$

ref: "#/components/responses/409Conflict"

# TOKEN\_REUSE|REFERRAL\_TENANT\_MISMATCH

4 22":

$

ref: "#/components/responses/422Unprocessable" # SELF\_REFERRAL\_BLOCKED|REFERRAL\_LIMIT

\_

REACHED

"

429":

$

ref: "#/components/responses/429RateLimited"

"

403":

description: Verboten oder Plan nicht erlaubt

content:

application/problem+json:

schema:

$ref: "#/components/schemas/Problem"

examples:

plan\_not\_allowed:

summary: Referral im Starter-Plan blockiert

value:

type: https://errors.lokaltreu.example/plan/not-allowed

title: Plan not allowed

status: 403

error\_code: PLAN\_NOT\_ALLOWED

correlation\_id: abc123

#

Plan-Gate bei Referral-Zweig MUSS 403 liefern. :contentReference[oaicite:8]{index=8}

"

500":

$ref: "#/components/responses/500ServerError"

/

rewards/redeem:

post:

tags: [Rewards]

summary: Prämie einlösen

description: Erfordert Device-Key und X-Device-Proof (Ed25519 über method|path|ts|jti). Id

empotent empfohlen. :contentReference[oaicite:9]{index=9}

security:

-

DeviceKey: []

parameters:

-

-

-

$ref: "#/components/parameters/XDeviceProof"

$ref: "#/components/parameters/XDeviceTimestamp"

$ref: "#/components/parameters/IdempotencyKey"

requestBody:

required: true

content:

application/json:

schema:

type: object

additionalProperties: false

required: [redeemToken]

properties:

redeemToken:

type: string

responses:

"

200":

description: Eingelöst

content:

application/json:

schema:

$ref: "#/components/schemas/RedeemResponse"

"

"

"

"

"

"

4 00":

r ef: "#/components/responses/400BadRequest" # TOKEN\_EXPIRED

4 01":

$

$

ref: "#/components/responses/401Unauthorized"

4 03":

$r ef: "#/components/responses/403Forbidden" # Proof ungültig

4 09":

$

ref: "#/components/responses/409Conflict"

# TOKEN\_REUSE

4 29":

$

ref: "#/components/responses/429RateLimited"

5 00":

$

ref: "#/components/responses/500ServerError"

/

referrals/link:

get:

tags: [Referrals]

summary: Personalisierter Referral-Link

description: Liefert refCode-URL für Werber. Plan-Gate aktiv. :contentReference[oaicite:10

{ index=10}

security: []

responses:

]

"

200":

description: OK

content:

application/json:

schema:

type: object

additionalProperties: false

required: [refCodeURL]

properties:

refCodeURL:

type: string

format: uri

"

403":

description: Plan nicht erlaubt

content:

application/problem+json:

schema:

$ref: "#/components/schemas/Problem"

examples:

plan\_not\_allowed:

summary: Starter-Plan blockiert Referral

value:

type: https://errors.lokaltreu.example/plan/not-allowed

title: Plan not allowed

status: 403

error\_code: PLAN\_NOT\_ALLOWED

correlation\_id: abc123

#

Referral-Routen prüfen Plan >= Plus. :contentReference[oaicite:11]{index=11}

"

"

429":

$

ref: "#/components/responses/429RateLimited"

500":

$

ref: "#/components/responses/500ServerError"

components:

securitySchemes:

AdminAuth:

type: http

scheme: bearer

bearerFormat: JWT

description: Admin-Access-Token (exp ≤ 15 Min). Refresh separat. :contentReference[oaicite

:

12]{index=12}

DeviceKey:

type: apiKey

in: header

name: X-Device-Key

description: Gerätegebundener API-Schlüssel.

headers:

Idempotency-Key:

description: >

Echo des Idempotenz-Schlüssels. Gültigkeit 24 h. Scope {tenantId,route,bodyHash}. :conte

ntReference[oaicite:13]{index=13}

required: false

schema:

type: string

minLength: 8

maxLength: 128

Retry-After:

description: Sekunden bis zum nächsten Versuch.

schema:

type: integer

minimum: 1

X-Device-Proof:

description: >

Ed25519-Signatur über (method|path|ts|jti), Base64url. Zeitdrift ±30 s zulässig. :conten

tReference[oaicite:14]{index=14}

schema:

type: string

parameters:

IdempotencyKey:

name: Idempotency-Key

in: header

required: true

description: Idempotenz für Schreibaktionen. 24 h gültig. :contentReference[oaicite:15]{in

dex=15}

schema:

type: string

minLength: 8

maxLength: 128

XDeviceProof:

name: X-Device-Proof

in: header

required: true

description: Ed25519-Signatur über (method|path|ts|jti). :contentReference[oaicite:16]{ind

ex=16}

schema:

type: string

XDeviceTimestamp:

name: X-Device-Timestamp

in: header

required: true

description: UNIX-Zeitstempel in Sekunden, für Signatur eingeschlossen. Skew ±30 s. :conte

ntReference[oaicite:17]{index=17}

schema:

type: integer

minimum: 0

responses:

4

00BadRequest:

description: Ungültige Anfrage

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

examples:

token\_expired:

summary: Token abgelaufen

value:

type: https://errors.lokaltreu.example/token/expired

title: Token expired

status: 400

error\_code: TOKEN\_EXPIRED

correlation\_id: abc123

4

4

4

0 1Unauthorized:

description: Nicht autorisiert

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

0 3Forbidden:

description: Verboten

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

0 9Conflict:

description: Konflikt

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

examples:

token\_reuse:

summary: Token bereits verwendet

value:

type: https://errors.lokaltreu.example/token/reuse

title: Token reuse

status: 409

error\_code: TOKEN\_REUSE

correlation\_id: abc123

4

22Unprocessable:

description: Semantisch fehlerhaft

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

examples:

self\_referral:

summary: Self-Referral blockiert

value:

type: https://errors.lokaltreu.example/referral/self

title: Self referral blocked

status: 422

error\_code: SELF\_REFERRAL\_BLOCKED

correlation\_id: abc123

4

29RateLimited:

description: Rate-Limit erreicht

headers:

Retry-After:

$

ref: "#/components/headers/Retry-After"

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

examples:

rate\_limited:

summary: Zu viele Anfragen

value:

type: https://errors.lokaltreu.example/rate/limited

title: Rate limited

status: 429

error\_code: RATE\_LIMITED

retry\_after: 10

correlation\_id: abc123

#

Limits: Tenant 600 rpm, IP 120 rpm, /stamps/claim 30 rpm/Card, /rewards/redeem 10 rpm/De

vice. :contentReference[oaicite:18]{index=18}

5

00ServerError:

description: Interner Serverfehler

content:

application/problem+json:

schema:

$

ref: "#/components/schemas/Problem"

schemas:

Problem:

type: object

additionalProperties: true

required: [type, title, status]

properties:

type:

type: string

format: uri

title:

type: string

status:

type: integer

detail:

type: string

instance:

type: string

error\_code:

type: string

enum:

-

-

-

-

-

-

-

TOKEN\_EXPIRED

TOKEN\_REUSE

SELF\_REFERRAL\_BLOCKED

REFERRAL\_LIMIT\_REACHED

REFERRAL\_TENANT\_MISMATCH

PLAN\_NOT\_ALLOWED

RATE\_LIMITED

#

Fehlercodes gem. Spezifikation. :contentReference[oaicite:19]{index=19}

correlation\_id:

type: string

retry\_after:

type: integer

minimum: 1

#

Domain-Modelle (Auszug aus ERD) :contentReference[oaicite:20]{index=20}

Tenant:

type: object

additionalProperties: false

required: [id, plan, createdAt]

properties:

id:

type: string

format: uuid

plan:

$

ref: "#/components/schemas/TenantPlan"

createdAt:

type: string

format: date-time

TenantPlan:

type: string

enum: [starter, plus, premium]

Campaign:

type: object

additionalProperties: false

required: [id, tenantId, stampsRequired, rewardLabel, active]

properties:

id:

type: string

format: uuid

tenantId:

type: string

format: uuid

stampsRequired:

type: integer

minimum: 1

rewardLabel:

type: string

minLength: 1

active:

type: boolean

description: "Constraint: je Tenant max. eine aktive Kampagne." # DB-Constraint. :contentR

eference[oaicite:21]{index=21}

StampToken:

type: object

additionalProperties: false

required: [jti, tenantId, deviceId, expiresAt]

properties:

jti:

type: string

tenantId:

type: string

format: uuid

deviceId:

type: string

format: uuid

expiresAt:

type: string

format: date-time

usedAt:

type: string

format: date-time

nullable: true

Referral:

type: object

additionalProperties: false

required: [code, tenantId, referrerCardId, referredCardId, qualified]

properties:

code:

type: string

tenantId:

type: string

format: uuid

referrerCardId:

type: string

referredCardId:

type: string

firstStampAt:

type: string

format: date-time

nullable: true

qualified:

type: boolean

bonusCreditedAt:

type: string

format: date-time

nullable: true

PlanCounter:

type: object

additionalProperties: false

required: [id, tenantId, month, stampsUsed, limit, devicesAllowed]

properties:

id:

type: string

format: uuid

tenantId:

type: string

format: uuid

month:

type: string

pattern: "^[0-9]{4}-[0-9]{2}$" # YYYY-MM

stampsUsed:

type: integer

minimum: 0

limit:

type: integer

minimum: 0

devicesAllowed:

type: integer

minimum: 0

#

API-spezifische Schemas

AdminRegistrationResponse:

type: object

additionalProperties: false

required: [adminId, tenantId, accessToken, refreshToken, expiresIn]

properties:

adminId:

type: string

format: uuid

tenantId:

type: string

format: uuid

accessToken:

type: string

refreshToken:

type: string

expiresIn:

type: integer

minimum: 1

DeviceRegistrationLinkResponse:

type: object

additionalProperties: false

required: [linkUrl, token, expiresAt]

properties:

linkUrl:

type: string

format: uri

token:

type: string

expiresAt:

type: string

format: date-time

qrImageUrl:

type: string

format: uri

nullable: true

StampTokenResponse:

type: object

additionalProperties: false

required: [qrToken, jti, expiresAt]

properties:

qrToken:

type: string

jti:

type: string

expiresAt:

type: string

format: date-time

#

siehe Sequenz 5.3. :contentReference[oaicite:22]{index=22}

StampClaimRequest:

type: object

additionalProperties: false

required: [qrToken]

properties:

qrToken:

type: string

ref:

type: string

nullable: true

CardState:

type: object

additionalProperties: false

required: [currentStamps, stampsRequired, rewardsAvailable]

properties:

currentStamps:

type: integer

minimum: 0

stampsRequired:

type: integer

minimum: 1

rewardsAvailable:

type: integer

minimum: 0

OfferSnippet:

type: object

additionalProperties: false

required: [title]

properties:

title:

type: string

body:

type: string

nullable: true

StampClaimResponse:

type: object

additionalProperties: false

required: [cardState]

properties:

cardState:

$

ref: "#/components/schemas/CardState"

offer:

$

ref: "#/components/schemas/OfferSnippet"

nullable: true

RedeemResponse:

type: object

additionalProperties: false

required: [cardState]

properties:

cardState:

$

ref: "#/components/schemas/CardState"

x-rateLimits:

tenant\_per\_minute: 600

ip\_anonymous\_per\_minute: 120

stamps\_claim\_per\_card\_per\_minute: 30

rewards\_redeem\_per\_device\_per\_minute: 10

#

Dokumentierte Limits. :contentReference[oaicite:23]{index=23}

`

``

Idempotenz nachgewiesen.

**1**

**2. PWA-Umsetzung**

Service Worker: stale-while-revalidate (statisch), network-first (API). Offline read-

only Kartenansicht. Manifest/Icons. Budgets: LCP ≤ 2.5 s, INP ≤ 200 ms, CLS ≤ 0.1.

**Verifizierbarkeit.** Lighthouse ≥ 90; „Installable: yes“.

**1**

**3. Geräteverwaltung**

Einmaliger Registrierungslink (TTL 15 Min), Bestätigungsseite, sofortige Invalidierung, E-

Mail-Alert bei Bindung, Sperren/Löschen.

**Verifizierbarkeit.** Confirm→Link unbrauchbar; gesperrtes Gerät→403.

**1**

**4. QR-gestützte Workflows & Anti-Replay**

**Token-Modell (MUSS).** StampToken(jti,expiresAt), RewardToken(jti,expiresAt).

**Anti-Replay (MUSS).** Redis SETNX(jti,"lock") → ACID-Transaktion (validate →

Geschäftsaktion → usedAt → Audit).

**QR-Parameter (MUSS).** ECC „Q“, ≥ 300×300 px, Kontrast ≥ 4.5:1; Scan p95 ≤ 2 s; E2E ≤ 3

s.

**Verifizierbarkeit.** Parallel-Tests (≥ 10 Threads) → exakt 1 Erfolg; Expiry → 400.



**1**

**5. Reporting & Dashboard**

**KPIs.** Stempel d/w/m, Rewards d/w/m, Geräteaktivität, Plan-Auslastung.

**Referral-KPIs.** referrals\_qualified\_\*, referral\_bonus\_stamps\_\*, Conversion „ref

clicks → first stamp“.

**Verifizierbarkeit.** SQL-Zeitreihen stimmen zu API-Metriken (< 1 % Abweichung).

**1**

**6. Preispläne & Limits**

**1**

**6.1 Plan-Regeln (MUSS)**

•

•

**Warnung bei 80 %**, **keine Blockade bei 100 %**. Upgrade jederzeit.

Geräteanzahl und Stempelkontingente planabhängig.

**1**

**6.2 Referral-Gate (NEU, MUSS)**

•

•

•

**Starter:** Referral **nicht aktivierbar**.

**Plus/Premium:** Referral aktivierbar.

**Backend-Guards (MUSS):** Plan-Check auf allen Referral-Routen (/referrals/\*,

Referral-Zweig in /stamps/claim). Bei Verstoß: 403 PLAN\_NOT\_ALLOWED.

**Admin-UI (MUSS):** Toggle nur sichtbar/aktiv bei Plan ≥ Plus; im Starter disabled mit

Hinweis.

•

•

**Negativtests (MUSS):** Starter→Block, Plus→OK, Downgrade→Auto-Disable.

**1**

**6.3 Plan-Limits ohne Sperre (NEU, MUSS)**

•

•

**Metriken:**

,

(once per 24 h), time-

plan\_usage\_percent plan\_warning\_emitted

to-upgrade-effective.

**Verhalten:** 80 %→Warnmail+Banner; 100 %→keine Blockade; Upgrade wirkt ohne

Downtime.

**Verifizierbarkeit.** Szenarien 79/80/100 % in Staging mit Messwerten.

**1**

**1**

**7. Admin-UI, Mitarbeiter-UI, Endkunden-PWA-UI**

•

**Admin.** Onboarding, Kampagne, Geräte, Reporting, Angebot, Plan/Upgrade, Referral-

Toggle (planabhängig).

•

•

**Mitarbeiter.** Zwei Buttons „Stempel vergeben“, „Prämie einlösen“, Auto-Return.

**Endkunde.** Karte, QR-Scan, Referral-Link.

**A11y-Hinweis.** Formale WCAG-Prüfung nicht Bestandteil; Verbesserungen geplant.

**8. Betrieb & Deployment**



**Plattform.** PaaS (EU-Region).

**CI/CD.** Build→Tests→Deploy (Blue-Green/Canary).

**Resilienz.** SLO 99,90 %, RPO 15 Min, RTO 60 Min; Multi-AZ; Degradationspfade:

Reporting read-only; Token-Erzeugung pausiert bei Redis-Partition; Claims für bereits

erzeugte Tokens erlaubt.

**JWKS.** Rotation 24 h Vorlauf; Rollback-Plan; Übungsprotokolle.

**Incident/Breach.** 24×7 On-Call; 72-h-Meldepfad (Art. 33/34).

**Verifizierbarkeit.** Probedurchläufe Rotation/Restore/Chaos dokumentiert.

**1**

**9. Teststrategie, Observability & Nachweise**

**1**

**9.1 Observability (MUSS)**

Metriken: p50/p95/p99 je Route; error\_5xx\_rate, rate\_token\_invalid,

rate\_token\_reuse, 429\_rate; Plan-Metriken (16.3). Traces: W3C TraceContext;

correlation\_id. Alerts: SLO-Breach, Fehlscan-Spike > 5/60 s/Tenant, Queue-Backlog.

**1**

**9.2 Nachweise & Testprotokolle (MUSS)**

•

•

•

**Geräte-Onboarding TTL/Einmaligkeit:** Expired→400, Reuse→409, E-Mail-Alert

beim Binden. Protokoll mit Zeitstempeln/Logs/Screenshots.

**Plan-Limits ohne Sperre:** 79/80/100 % Messbericht inkl. plan\_usage\_percent,

plan\_warning\_emitted, time-to-upgrade-effective.

**Referral-Gate:** Starter→403 PLAN\_NOT\_ALLOWED, Plus→200, Downgrade→Auto-

Disable belegt.

•

•

**Loadtests Hot-Routes:** /

**Restore/Chaos vs. RPO/RTO.**

stamps/claim /rewards/redeem

p50/p95/p99.

,

**1**

**9.3 Coverage (MUSS)**

Unit ≥ 80 %, Integration, E2E; Contract-Tests; Parallel-Redeem; Referral-Grenzfälle.

**Verifizierbarkeit.** CI verlinkt Berichte; rote Alerts testweise ausgelöst.

**2**

**0. Risiken & Annahmen**

•

•

•

**Zeitdrift** → NTP-Monitoring, Skew ± 30 s.

**Key-Rotation-Fehler** → geübtes Runbook & Rollback.

**Übernutzung** → keine Blockade, frühe Warnungen.

**Verifizierbarkeit.** Runbooks vorhanden und geübt.



**2**

**1. Rückverfolgbarkeitsmatrix (Auszug)**

**Proz**

**ess**

**US**

**APIs**

**Datenobjekte**

**UI**

**Tests**

**Logs/Metriken**

Mitarb

eiter

/

stamps/tokens

rate\_token\_re

8

1

A

,

TC-A-Parallel

StampToken Stamp

, /stamps/claim

use, p95\_claim

/

m

rewards/redee RewardToken

1 B

PWA TC-B-Expiry p95\_redeem

Reward

/

referrals/lin Referral,

Stamp(meta=referr

stamps/claim al\_bonus)

referrals\_qua

lified\_d

1

3 5.5

,

PWA TC-RF-01..04

k

/

plan\_warning\_

Admin-Plan-

APIs, Guards

Referral-Routen Tenant.plan

,

TC-PLAN-

Admin Starter/Plus/Do

wngrade

PlanCounter

Pla 5.1/1

,

Feature-Flag,

emitted time-

to-upgrade-

effective

n

6

/

devices/regis

security\_emai

De 5.2/1

Admin/ TC-DEV-

tration-links, Device, AuditLog

l\_sent,

v

3

Dev

Expired/Reuse

confirm

4

09\_rate

TC-Scope-

Admin Parallel-

Campaign

Constraint-

Violations

(DB+API)

Sco

pe

5

.1 /campaigns/\*

Campaign

**2**

**2. Glossar**

**ACID** • **DPoP** • **JTI** • **JWKS** • **PWA** • **SLO** • **RPO/RTO** • **WORM** • **RoPA** • **TOMs** •

**DSR**.

**2**

**3. Anhänge (MUSS)**

•

•

**Compliance:**

Infos-DE.pdf

Anhänge/Compliance/AVV.pdf TOMs.pdf RoPA.pdf DPIA.pdf

,

,

,

,

**Runbooks:** Anhänge/Runbooks/JWKS-Rotation.md, JWKS-Rollback.md,

Incident-Breach.md, Restore.md, Replay-Suspected.md

•

•

**Reports:**

,

Anhänge/Reports/Loadtest-stamps-claim.pdf Loadtest-rewards-

redeem.pdf, Restore-RPO-RTO.pdf, Chaos-Resilience.pdf

**API-Bundle:** Anhänge/API/lokaltreu-openapi-v2.0.yaml

**Normative Ausschlüsse (MUSS)**

•

•

•

Keine monetären Auszahlungen.

Keine Zahlungsdienstleister.

Keine Payout-Integrationen.



**Ergebnis.** Die Spezifikation ist verständlich, auditierbar und Go-Live-tauglich. Alle

geforderten Präzisierungen sind verbindlich integriert.

**Abschnitt 3**

Quelle: Architektur-Lokaltreu.pdf

*Hinweis: Originalinhalte folgen unverändert im Anschluss an diese Abschnittsseite.*

Empfehlung: **Modularer Monolith auf EU-PaaS**. TypeScript durchgängig. PostgreSQL +

Redis. CDN + EU-Object-Storage. Transaktionsmails in EU. Ergebnis: geringe Fixkosten,

einfache Wartung, klarer Skalierungspfad. Architektur und Dienste unten.

Architektur

•

•

**Stil:** Modularer Monolith auf PaaS. Module: Auth, Devices, QR/Tokens,

Stamps/Rewards, Referral, Reporting, Admin.

1

Technische Spezifikation – Lok…

**Kernpfade:** QR-Token mit jti+TTL, Anti-Replay via Redis **SETNX**, ACID-

Transaktion, unveränderliches Audit.

1

Technische Spezifikation – Lok…

**NFRs:** p95 ≤ 3 s, SLO 99,9 %, RPO 15 min, RTO 60 min, Multi-AZ.

Technische Spezifikation – Lok…

**PWA:** SW „stale-while-revalidate“ (statisch), „network-first“ (API), LCP ≤ 2,5 s.

Technische Spezifikation – Lok…

•

•

1

1

Technologie-Stack

•

**Frontend:** Next.js + React + TypeScript + Tailwind. PWA-First.

1

Technische Spezifikation – Lok…

•

**Backend:** Node.js + TypeScript, REST/JSON, OpenAPI 3.1, Fehler RFC 7807,

Idempotency-Key.

1

Technische Spezifikation – Lok…

•

**Security:** Admin-JWT (Access ≤ 15 min, Refresh ≤ 30 d), Geräte Ed25519 + **X-**

**Device-Proof**; Endkunden anonym. Rate-Limits pro Tenant/IP/Card/Device.

1

Technische Spezifikation – Lok…

**Datenbank:** PostgreSQL mit Constraints (u. a. „eine aktive Kampagne je Mandant“).

Technische Spezifikation – Lok…

**Cache/Queues:** Redis für Locks, Rate-Limits, kurze Jobs.

Technische Spezifikation – Lok…

**Audit:** WORM-Log, 180 Tage, signierte Exporte.

•

•

•

1

1

1

Technische Spezifikation – Lok…

Konkrete Dienste (EU-Regionen)

**App-Hosting**

•

**Primär:** Fly.io (eu-central) oder Render.com (Frankfurt). PaaS, Auto-Scaling, Blue-

Green/Canary. Einsatz: API-Monolith, leichte Worker.

1

Technische Spezifikation – Lok…

**Datenbank (PostgreSQL)**

•

**Primär: Neon** (EU). Serverless Branching für Staging/Previews. Einsatz: Tenants,

Campaigns, Stamps, Rewards, Referrals, PlanCounter.

1

Technische Spezifikation – Lok…

**Redis/Cache**

•

**Primär: Upstash Redis** (EU). Nutzung: Anti-Replay **SETNX(jti)**, Idempotenz-

Locks, Rate-Limits, kleine Queues.

1

Technische Spezifikation – Lok…

**Object Storage**

•

**Primär: Cloudflare R2** mit EU-Jurisdiction. Nutzung: signierte Audit-Exporte

(WORM-Snapshots), Reporting-Dumps, Medien/Assets.

1

Technische Spezifikation – Lok…

**CDN**

•

**Primär: Cloudflare CDN** mit **Regional Services** (EU-TLS-Terminierung). Nutzung:

PWA-Delivery, Static/Images, Edge-DDoS-Schutz.

1

Technische Spezifikation – Lok…

**Mail-Service**

•

**Primär: Mailjet** oder **Brevo** (EU-Datenhaltung, DPA). Nutzung: Sicherheits-Alerts

(Gerätebindung), Plan-Warnungen 80 %, Admin-Einladungen.

Zuordnung „Lokaltreu“ → Plattformen

•

**QR-Validierung / Stempel-Claim (/stamps/claim)** → App-Hosting (Fly/Render) +

Redis (Anti-Replay) + Postgres (Transaktion) + Audit-Append.

•

**Prämieneinlösung (/rewards/redeem)** → App-Hosting + Redis-Limits + Postgres.

1

Technische Spezifikation – Lok…

•

•

**Referral-Gate + Server-Guards** → App-Hosting + Postgres + Feature-Flags; Plan-

Fehler 403 **PLAN\_NOT\_ALLOWED**.

1

Technische Spezifikation – Lok…

**Admin-Onboarding & Gerätebindung** → App-Hosting + Mail-Service (Security-

Alert) + Audit.

1

Technische Spezifikation – Lok…

**Reporting/Dashboard** → Postgres (Zeitreihen), periodische Exporte nach R2.

Technische Spezifikation – Lok…

**PWA-Delivery** → Cloudflare CDN + R2 für Assets.

Technische Spezifikation – Lok…

•

•

•

1

1

**Unveränderliches Audit** → Postgres WORM-Tabelle + periodische signierte R2-

Exports (180 Tage).

1

Technische Spezifikation – Lok…

Betriebsmodell

•

**PaaS in EU:** Private Netze, TLS erzwungen, Multi-AZ, Auto-Scaling, Blue-

Green/Canary.

1

Technische Spezifikation – Lok…

**Environments:** Dev/Stage/Prod, OpenAPI-Contracts, CI/CD-Gates.

Technische Spezifikation – Lok…

•

•

1

**Observability:** OpenTelemetry Metriken p50/p95/p99 je Route, 5xx-Rate,

rate\_token\_reuse/invalid, 429-Rate; Alerts bei SLO-Breach und Fehlscan-Spikes.

1

Technische Spezifikation – Lok…

•

**Runbooks:** JWKS-Rotation/Rollback, Restore, Incident/Breach, Replay-Verdacht.

Übungsprotokolle.

1

Technische Spezifikation – Lok…

DSGVO, Auditierbarkeit, Anti-Abuse

•

•

•

**Rechtsgrundlage:** Art. 6 Abs. 1 lit. f für Betrieb/Sicherheit/Fraud-Prevention.

Endkunden anonym, pseudonyme Card-IDs.

1

Technische Spezifikation – Lok…

**Logs als personenbezogene Daten:** RoPA-Eintrag, TOMs, Löschautomatik; DSR

nach Art. 11 ohne zusätzliche Identifizierung.

1

Technische Spezifikation – Lok…

**Consent:** nur technisch notwendige LocalStorage/Cookies, kein Banner nötig; Info im

Datenschutzhinweis.

1

Technische Spezifikation – Lok…

•

•

**Plan-Limits:** Warnung bei 80 %, keine Blockade bei 100 %, sofortiges Upgrade

möglich; UI-Banner + Mail.

**Anti-Abuse:** strikte TTL, jti-Einmaligkeit, Velocity-Limits, Mandantenbindung, Self-

Referral-Block.

1

Technische Spezifikation – Lok…

Kostenübersicht (MVP: ~100 Händler, ~40

000 Endkunden)

•

**Hosting (Fly/Render):** 1–2 App-Instanzen + 1 Worker-Instanz. Lastspitzen durch

CDN abgefangen. p95-Ziel bleibt erreichbar. Fixkosten moderat, nutzungsabhängige

Skalierung.

1

Technische Spezifikation – Lok…

•

•

•

**Postgres (Neon):** kleine produktionsfähige EU-Instanz, Branches für Staging ohne

Duplikatkosten.

**Redis (Upstash):** nutzungsbasiert; Anti-Replay/Limits verursachen kurze

Operationen.

**Object-Storage (R2):** niedrige Speicherkosten, praktisch kein Egress für interne

Audits.

•

•

**CDN (Cloudflare):** reduziert Origin-Last für PWA/Assets.

**Mail (Mailjet/Brevo):** nur transaktionale Mengen (Alerts, Plan-Warnungen).

**Hebel:** Caching aggressiv, kurze TTLs, schlanke Logs (180 Tage), Reporting batchen,

Staging ressourcensparend. **Keine** Endkunden-Mailings im Scope.

1

SaaS-Beschreibung

Skalierungspfad (EU-weite Expansion)

1

. **Horizontal skalieren:** mehr App-Instanzen; Redis- und Postgres-Pläne hochstufen.

Multi-AZ bleibt.

1

Technische Spezifikation – Lok…

2

3

. **Worker entkoppeln:** Reporting/Email-Jobs als separate Worker-Apps.

. **Read-Replicas/Partitionierung:** Audit/Events partitionieren; Reporting gegen

Replica.

1

Technische Spezifikation – Lok…

4

5

. **Hotspot-Abspaltung bei Bedarf:** nur eng begrenzte Domänen (z. B. /stamps/claim)

als Service abtrennen, interne Auth beibehalten.

. **Edge-Optimierung:** CDN-Caching für statische API-Responses, wenn unverfänglich.

Wartbarkeit für kleine Teams

•

**Monorepo:** Frontend, Backend, IaC. Gemeinsame Typen aus OpenAPI.

1

Technische Spezifikation – Lok…

•

**Qualität:** Contract-Tests, Parallel-Tests für Replay-Szenarien, Coverage-Ziele, SLO-

Dashboards in CI verlinkt.

1

Technische Spezifikation – Lok…

**Feature-Flags/Plan-Gates:** serverseitig prüfbar, UI-Toggle planabhängig.

Technische Spezifikation – Lok…

•

1

Zusammenfassung der Plattform-

Zuordnung

•

•

•

•

•

**Fly.io/Render** → API-Monolith, Worker, Blue-Green/Canary.

Technische Spezifikation – Lok…

**Neon (Postgres)** → Kernobjekte, Constraints, Transaktionen.

Technische Spezifikation – Lok…

**Upstash (Redis)** → Anti-Replay, Idempotenz-Locks, Rate-Limits, kleine Queues.

Technische Spezifikation – Lok…

**Cloudflare R2** → Audit-Exports (WORM, 180 Tage), Reports, Medien.

Technische Spezifikation – Lok…

**Cloudflare CDN** → PWA-Delivery, Edge-Schutz, EU-Regional Services.

1

1

1

1

1

Technische Spezifikation – Lok…

•

**Mailjet/Brevo** → Security-Alerts, Plan-Warnungen, Admin-Einladungen.

Diese Konfiguration erfüllt die Spezifikation, bleibt DSGVO-konform, ist auditierbar und

skaliert mit geringer Betriebslast.