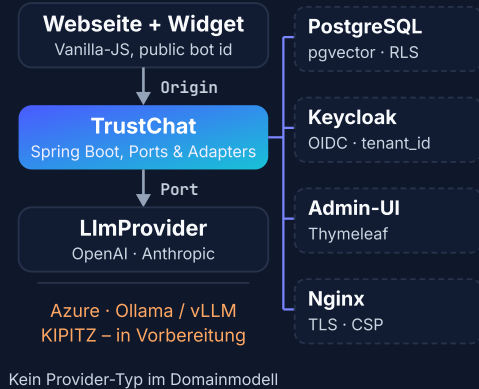


1 Architektur



2 Funktionen

Private by Default

PostgreSQL Row-Level Security, FORCE RLS, SET LOCAL je Transaktion

Model Adapters

LLM-Zugriff nur über einen Port – Provider austauschbar, gleicher Code

Quotas

Tages- und Monatslimits je Tenant, Überlauf wird sauber blockiert

Spezialregeln

Deterministische Q&A-Overrides je Bot, mit Vorrang vor der Wissensbasis

Self-Hosted

Spring Boot · PostgreSQL · Keycloak · Nginx – keine SaaS-Abhängigkeit

RAG & Wissensbasis

MD · TXT · PDF · URLs, hybrid (pgvector + Volltext), je tenant_id/bot_id

Observability

Prometheus-Metriken, Usage-Events, Kosten je Modell, Logs ohne Secrets

SSE Streaming

Antworten live per SSE: Widget, eigene Bot-Seite und Admin-Chat

3 Widget einbetten

Eine Zeile. Auf jeder Webseite.

```
<script src="https://<host>/widget.js" data-bot-id="public_<hex>"></script>
```

- ✓ Snippet unter /admin/deploy, Farbe per data-accent; alternativ /c/<id>
- ✓ Domain muss in der Allowlist des Bots stehen (serverseitig)
- ✓ Rate Limit je Besucher (Standard 30 Anfragen/min)
- ✓ Browser sieht nie Tenant-ID, JWT, API-Key oder Bot-UUID

4 In fünf Schritten zum eigenen Bot

01 Anmelden

Admin-Login über Keycloak (OIDC)

02 Bot anlegen

Name, Modell und public bot id

03 Wissen laden

Dateien und URLs, dann trainieren

04 Regeln setzen

Spezialregeln und Domain-Allowlist

05 Einbetten

Widget-Zeile oder eigene Bot-Seite

5 KI-Modelle

- OpenAI GPT-6 Luna · Sol · Astra verfügbar
 - Anthropic Claude 5.5 · Fable verfügbar
 - Azure OpenAI in Vorbereitung
 - Ollama / vLLM (lokal) in Vorbereitung
 - KIPITZ in Vorbereitung
- Pro Bot wählbar, über den Provider-Adapter.

6 Editionen

Community

1 Bot · 10 Nachrichten/Tag
100 Nachrichten/Monat

€0 / Monat

Pro

500/Tag · 10.000/Monat, Allowlist, Quellen, Leads, Chatverläufe

Einzellizenz

Enterprise

Mehrere Bots, eigene Provider, dedizierte Instanz, SLA/AVV/Audit

individuell

7 Sicherheit

- ✓ **Tenant-ID** nur aus JWT-Claim tenant_id
- ✓ **RLS** FORCE, Rolle NOBYPASSRLS
- ✓ **Cross-Tenant** Tests sind Pflicht
- ✓ **Allowlist** Domain je Bot, serverseitig
- ✓ **CSP / CSRF** kein Inline-JS im Admin
- ✓ **Log-Hygiene** keine Keys und JWTs im Log
- ✓ **BYOK** Tenant-Keys mit AES-GCM

8 Self-Hosting

Java 25 · Spring Boot 4
Datenbank PostgreSQL 18 + pgvector
Identität Keycloak 26 (OIDC)
Betrieb Docker Compose · Nginx

- 100 % self-hosted
- Linux-first & Docker-ready
- Air-gapped Deployments möglich
- Compliance-freundlich by design

9 Stand und Kontakt

Phase MVP im Produktivbetrieb
Tests Unit, Integration, ArchUnit
Gate JaCoCo ab 80 % Zeilen/Zweige

Grenzen: Rate Limit je Knoten, DSGVO-Audit folgt in Phase 2

trustchat.de

info@trustchat.de

BENENNUNG	VERSION	STAND
Cheat Sheet	0.1.0	19.10.2026