ARCHITEKTUR & KI-STRATEGIE

Skalierbare KI für Multisite-Systeme: Effiziente RAG-Architekturen mit Chroma DB Tenants

Von Michael Kettel | rms. Relationship Marketing Solutions GmbH

Wer große Portal-Netzwerke, Corporate Multisites oder verzweigte Systemlandschaften betreut, steht bei der Integration moderner KI-Features vor einer massiven Herausforderung: Wie bringt man intelligente Suche, semantische Filter und RAG-Systeme (Retrieval-Augmented Generation) auf Hunderte von Subportalen, ohne im Infrastruktur-Chaos zu versinken? Die Antwort liegt in einer nativen Mandantenfähigkeit. Ein tieferer Blick auf das „Tenant“-Konzept von Chroma DB zeigt, wie moderne Multisite-Umgebungen architektonisch sauber aufgestellt werden.

Die Herausforderung: KI-Suche im Multisite-Dilemma

Ob Enterprise-Kunde mit spezialisierten Marken-Subsites oder Hauptportale mit regionalen Ablegern: Die Inhaltsstruktur ist klar separiert. Das Marketing-Subportal benötigt keinen Zugriff auf die technischen Dokumentationen des Service-Portals. Dennoch soll nicht für jedes einzelne Subportal ein eigener, teurer KI-Tech-Stack oder eine separate Datenbank-Instanz hochgezogen werden.

Bisherige Ansätze lösten dies oft über komplexe Metadaten-Filter innerhalb einer einzigen großen Vektorsammlung (Collection). Doch das rächt sich schnell:

- **Sicherheitsrisiken:** Datenlecks zwischen Portalen durch fehlerhafte Filter-Logiken in der Applikationsschicht.
- **Performance-Einbußen:** Je größer die gemeinsame Collection wird, desto träger und fehleranfälliger werden die semantischen Suchanfragen.
- **Wartungs-Overhead:** Die saubere Trennung bei Updates, Löschungen oder Re-Indexierungen einzelner Subportale wird zum logistischen Albtraum.

Die Lösung: Datenisolierung auf Datenbank-Ebene durch „Tenants“

Hier kommt das native Multi-Tenancy-Konzept von **Chroma DB** ins Spiel. Im Code und in der API als `Tenants` (engl. für Mieter/Mandanten) bezeichnet, erlaubt es dieses Feature, eine einzige Chroma-Instanz zentral zu betreiben und die Daten dennoch vollkommen strikt voneinander zu trennen.

Chroma DB baut dafür eine klare, dreistufige Hierarchie auf:

- **1. Tenant (Mandant):** Die oberste logische Klammer – idealerweise das jeweilige Subportal oder der eigenständige Themenbereich.

- **2. Database (Datenbank):** Ein Tenant kann wiederum mehrere logische Datenbanken besitzen (z. B. zur Trennung von `Live-Inhalten` und `Staging/Testing`).
- **3. Collection (Sammlung):** Die eigentliche Ebene, auf der die Vektoren, Texte und Einbettungen (Embeddings) für die semantische Suche leben.

Architektonische Vorteile für Multisite-Betreiber

Durch diese saubere Kapselung ergeben sich entscheidende Vorteile für die Entwicklung und den Betrieb von modernen KI-gestützten CMS- und Portal-Umgebungen:

1. Absolute Datensicherheit ohne Programmieraufwand

Die Kapselung erfolgt direkt im Core der Vektordatenbank. Ein Client, der auf den Tenant eines spezifischen Subportals konfiguriert ist, kann physikalisch nicht auf die Daten eines anderen Tenants zugreifen. Programmierfehler in der Filter-Logik der Applikation führen somit niemals zu mandantenübergreifenden Datenlecks.

2. Unabhängige Indexierung und Updates

Wird auf einem Subportal ein umfassender Relaunch durchgeführt oder Content massiv geändert, betrifft die Neuindexierung der Vektoren ausschließlich diesen einen Tenant. Das Hauptportal und alle anderen Subseiten laufen ungestört mit maximaler Performance weiter.

3. Ressourceneffizienz und einfaches Hosting

Anstatt hunderte kleine Datenbank-Container zu orchestrieren, wird ein zentraler, leistungsfähiger Chroma-Cluster betrieben. Das spart drastisch Infrastrukturkosten und vereinfacht das Monitoring sowie Backups erheblich.

Praxis-Einblick: So einfach steuert die Anwendung die Mandanten

Die programmatische Umsetzung ist dank der Admin-API von Chroma DB extrem schlank. Wenn ein neues Subportal im CMS provisioniert wird, legt das System im Hintergrund vollautomatisch die neuen, isolierten Strukturen an:

```
Beispiel: Dynamische Mandantentrennung in Python
```

```
import chromadb

# 1. Zentraler AdminClient erstellt die isolierte Struktur für ein neues Subportal
admin_client = chromadb.AdminClient(host="chroma.internal.network", port=8000)
admin_client.create_tenant(name="subportal_e_mobilitaet")
admin_client.create_database(name="content_prod", tenant="subportal_e_mobilitaet")

# 2. Das Subportal verbindet sich exakt nur mit seinem eigenen geschützten Bereich
subportal_client = chromadb.HttpClient(
    host="chroma.internal.network",
    port=8000,
    tenant="subportal_e_mobilitaet",
    database="content_prod"
)

# Diese Collection ist für das Hauptportal oder andere Subseiten unsichtbar
collection = subportal_client.get_or_create_collection(name="knowledge_base")
```

Fazit: Die zukunftssichere Basis für Corporate AI

Die Integration von künstlicher Intelligenz in komplexe Web-Plattformen darf nicht zu einem unüberschaubaren Flickenteppich an Systemen führen. Das Tenant-Konzept von Chroma DB zeigt, wie moderne Softwarearchitektur aussehen muss: Zentral administrierbar, ressourcenschonend im Hosting, aber absolut strikt und sicher in der logischen Datentrennung. Für Betreiber von komplexen Multisite-Systemen ist dieser Ansatz der Schlüssel, um KI-Features wie semantische Suchen oder automatisierte Redaktions-Assistenten skalierbar und datenschutzkonform auf die Straße zu bringen.

Über rms. Stuttgart

rms. Relationship Marketing Solutions GmbH ist Ihre Digital- und IT-Agentur für zukunftssichere Ökosysteme. Mit über 20 Jahren Erfahrung in der Realisierung komplexer Enterprise-Plattformen begleiten wir Unternehmen strategisch und technologisch bei der Digitalisierung. Unser Fokus liegt auf der nahtlosen und tiefen Integration von Künstlicher Intelligenz (KI), intelligenten RAG-Architekturen und autonomen Systemen in bestehende IT- und Marketing-Infrastrukturen – für skalierbare Lösungen, die höchste Ansprüche an Datensicherheit und Performance erfüllen.