

So entstand ruum. SaaS mit vibe-coding



Whitepaper · Stand September 2026 · Für Website und Kunden.
Warum und wie Montrel ruum gebaut hat, vom Proof-of-Concept bis zum laufenden Betrieb.

Warum wir ruum gebaut haben

Anfang 2026 fanden wir auf dem Markt keine Lösung, die für Schweizer Gemeinden wirklich passte. Es gab Kalender. Es gab Einzeltools. Was fehlte, war der ganze Raumbetrieb: Anfragen, Freigaben, Preise, Sperrzeiten, Rollen, Facility. Also genau das, was eine Verwaltung im Alltag braucht.

Gleichzeitig war KI plötzlich brauchbar genug, um einen ernsthaften Versuch zu wagen. Wir wollten nicht drei Jahre Konzept schreiben. Wir haben einen Proof-of-Concept gemacht. Frage: Lässt sich mit AI (vibe-coding) und festen Regeln ein Multi-Tenant-SaaS bauen, dem eine Gemeinde vertrauen kann?

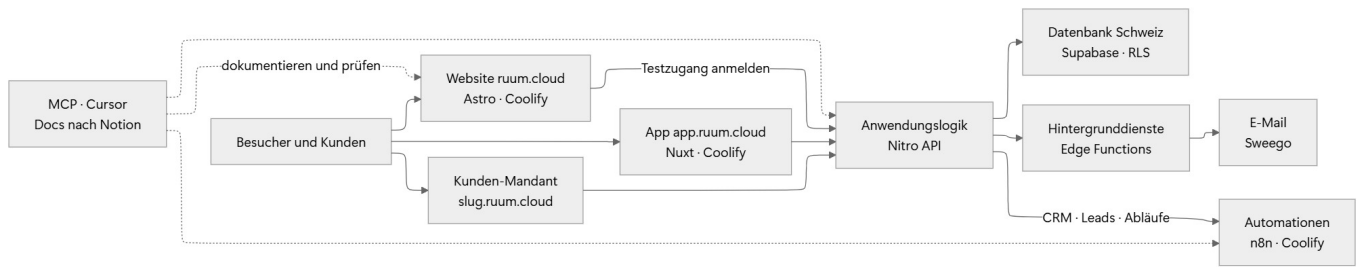
Aus dem PoC wurde ruum. Testzugang, Vertrag, Alltag, Archivierung. Sicherheit und Dokumentation sind keine Nachgedanken. Sie stehen im Ablauf.

Vibe-coding heisst: Software mit KI-Hilfe schreiben. Das ist schnell. Ohne Regeln wird es unsauber. Bei uns schreibt die KI mit. Freigabe, Checks und Docs entscheiden, was live geht.

1. Woraus ruum besteht

Die Website erklärt das Produkt und startet den 14-Tage-Test. Die App ist das Arbeitswerkzeug für Verwaltung und Buchung. Beides läuft auf Coolify, unserem eigenen Hosting. Die Datenbank steht in der Schweiz. Mails und Hintergrund-Automationen hängen nicht am kritischen Buchungspfad.

MCP (Model Context Protocol) verbindet unseren KI-Arbeitsplatz Cursor mit Notion, der Datenbank und Automationen. So landet Wissen in Notion, nicht nur im Chat.



- Website: ruum.cloud, Astro auf Coolify.
- App: Buchung und Verwaltung unter `app.ruum.cloud` und `{slug}.ruum.cloud`, Nuxt 4, Vue 3.
- Datenbank: Jede Organisation hat einen eigenen Mandanten. Row Level Security erzwingt die Trennung. Standort Zürich, Supabase, PostgreSQL.
- E-Mail: System- und Buchungsmails über Sweego (EU), Absender `notify.ruum.cloud`.
- n8n: Automationen für CRM und Leads. Nicht der Hotpath einer einzelnen Buchung, Coolify.
- MCP: Betriebsentscheidungen und Docs landen im Knowledge Hub auf Notion, im gleichen Arbeitsfluss wie Code und Checks sowie im Changelog ruum.cloud/changelog
- Schlüssel: Im Browser nur öffentliche Zugangsdaten. Geheime Schlüssel bleiben auf dem Server.

2. So entwickeln wir

Zuerst zeichnen wir und schreiben auf, was gebraucht wird. Kein Code auf Verdacht. Dann klären wir Sicherheit: Wer darf was? Sind Mandanten getrennt? Wo liegen Geheimnisse? Welche Datenbankänderungen stehen an?

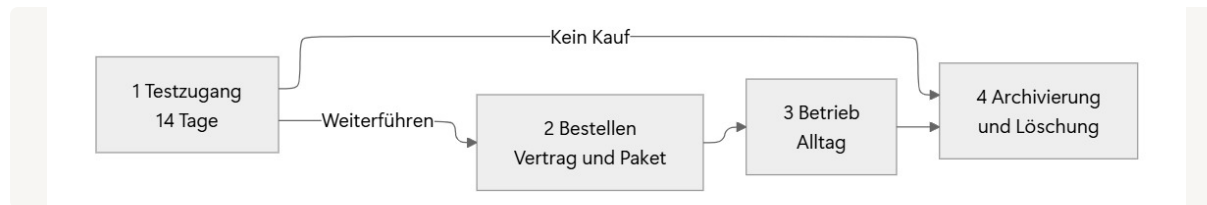
Erst danach baut die KI unter festen Projektregeln (Cursor mit verbindliche Anweisungen wie Agents, Skills und Rules).

Danach laufen automatische Checks (Workflows): Security, Stil, Typen, Tests, die prüfen z.B., dass Mandanten sich nicht gegenseitig sehen. Danach die Testumgebung mit komplett getrennten Daten von Produktion `preview.ruum.cloud`, menschliche Freigabe, dann Produktion.

Zum Schluss dokumentiert der Agent über MCP nach Notion und den Changelogs. Auffindbar fürs Team, für Audits für Kunden = volle Transparenz. Wichtig ist zu verstehen, die KI macht Tempo, sie ist ein Werkzeug. Menschen machen Checks und entscheiden, was live geht.

3. Der Weg eines Kunden

Von der Anmeldung bis zur Löschung der Daten:



1. **Testzugang:** Auf ruum.cloud anmelden. Eigener Mandant für 14 Tage, Willkommens-E-Mail, Erinnerungen sieben und drei Tage vor Ablauf. Ohne Weiterführung pausieren wir den Zugang.
2. **Bestellen:** Angebot und Vertrag (AGB, Auftragsbearbeitung), Paket Basis, Standard oder Enterprise. Ziel für die Einrichtung: innert 30 Arbeitstagen, wenn die Organisation mitzieht. Online-Selbstkasse ist bewusst noch nicht der Normalfall. Beratung und Vertrag bleiben zentral.
3. **Betrieb:** Alltag: Anfragen, Freigaben, Preise, Sperrzeiten, Rollen in Verwaltung und Facility. System-E-Mails informieren. Verfügbarkeitsziel: 99,0 % im Monat, ohne geplante Wartung.
4. **Archivierung:** Daten bleiben nicht ewig aktiv. Zuerst Archiv, dann Löschstufen gemäss Einstellungen. Nach Vertragsende: Export auf Wunsch innert 30 Tagen, Löschung des Mandanten innert 90 Tagen, gesetzliche Aufbewahrung vorbehalten. Ungenutzte Trials bereinigen wir nach einer Schonfrist.

4. Sicherheit, kurz gesagt

Jede Organisation hat ihren eigenen Bereich. Die Datenbank lässt keinen Blick auf fremde Mandanten zu. Rechte folgen «nur so viel wie nötig». Verbindungen sind verschlüsselt. Passwörter speichern wir nie im Klartext. App und Website laufen auf Coolify in DE. Die Datenbank mit den Kundendaten steht in der Schweiz.

5. Was als Nächstes gilt

Jeder Release geht denselben Weg: klären, absichern, bauen, prüfen, dokumentieren, freigeben. Die KI spart sehr viel Zeit. Prozess und Tests sind aber dennoch wichtig und halten die Qualität. Ändert sich Hosting oder ein Partner, ziehen wir TOMs und Betriebsunterlagen nach. Kunden sollen nachvollziehen können, wie ruum läuft.

6. Learnings aus dem PoC

Was wir aus der Entwicklungen von ruum mitnehmen:

1. Der KI nicht blind vertrauen. Die KI liefert Vorschläge. Verantwortlich bleibt ein Mensch.
2. Immer nachdenken und -prüfen. Fertig ist erst, wenn Checks und gezielte Tests grün sind.
3. Sicherheit zuerst denken. Rechte, Mandantentrennung und Geheimnisse klären, bevor jemand Code schreibt. Nicht erst nach dem Vorfall.
4. Feste Regeln mit Skills und Rules statt freier Chat. Ohne Leitplanken kopiert KI unsichere Muster. Projektregeln halten den Rahmen.
5. Klein und an der Ursache fixen. Lieber eine saubere Korrektur an der richtigen Stelle als viele Schnellschüsse. Bei grösserem Change zuerst das Konzept, dann der Code.

Testzugang und Infos: ruum.cloud. Fragen: support@montrel.org.

Herausgeber: Verein Montrel. Produkt: ruum. Stand: September 2026.