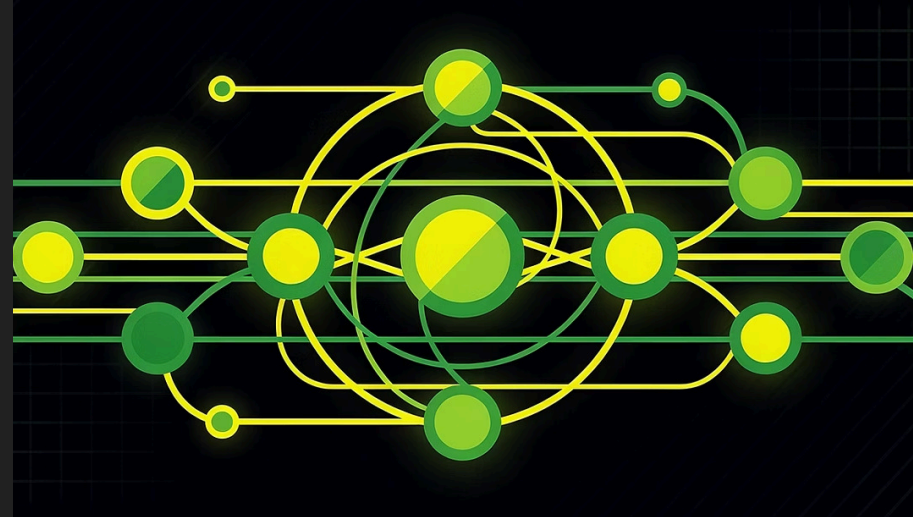
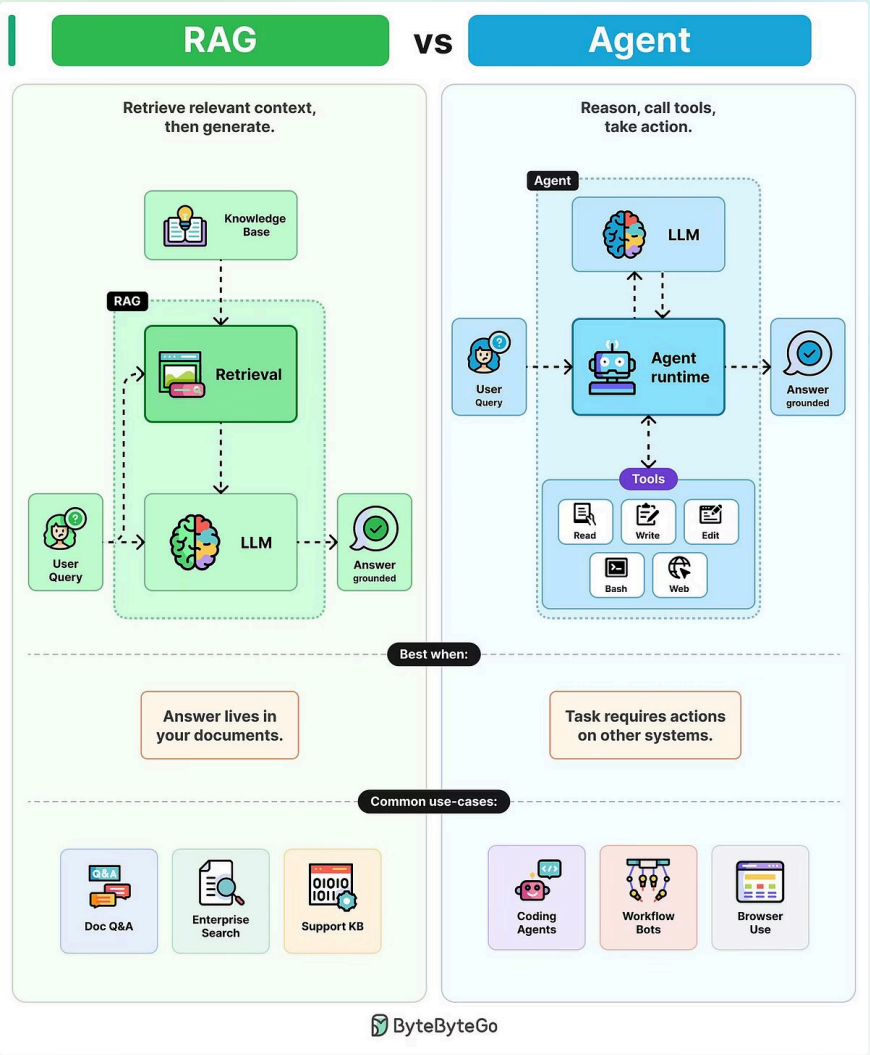
RAGs vs Agenti

Opýtajte sa LLM na dáta vašej spoločnosti a bude hádať. Dva vzory, ktoré to riešia, sú RAG a agenti — a každý rieši iný problém.

BYTEBYTEGO · 23. MÁJA 2026





RAGs vs Agenti: Prehľad

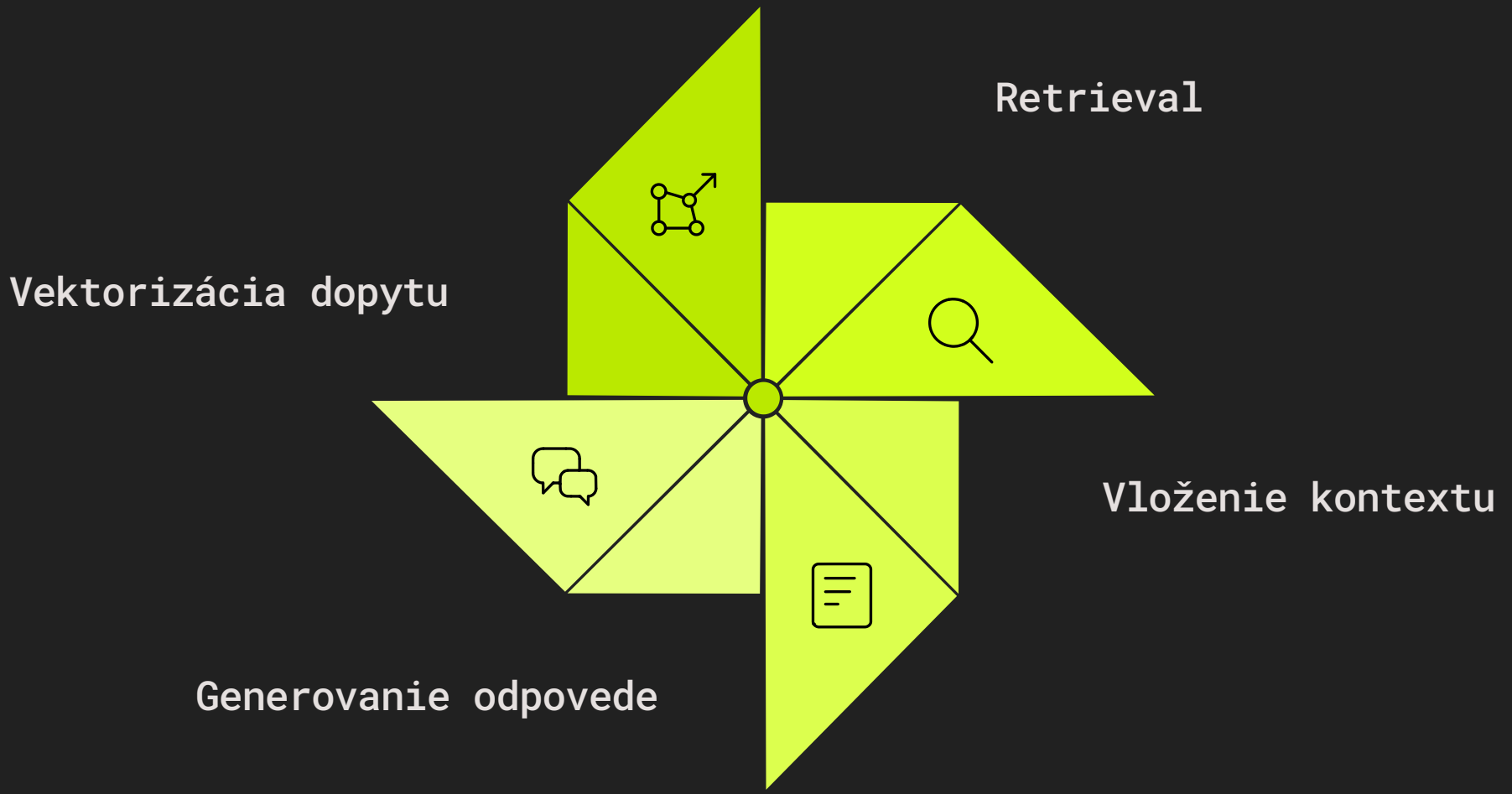
Kedy použiť RAG

Keď odpoveď žije vo vašich dokumentoch — doc Q&A, podnikové vyhľadávanie, znalostná báza podpory.

Kedy použiť Agenta

Keď úloha vyžaduje akcie na iných systémoch — kódovacie agenty, workflow boty, ovládanie prehliadača.

Ako funguje RAG



Jedno získanie. Jedna generácia. **Lacné, predvídateľné a ľahko laditeľné.** RAG kombinuje LLM s retrieval krokom, aby zakotvil odpovede v reálnych dokumentoch.

Ako fungujú Agenti

Príjem dopytu

LLM vyberie nástroj

LLM uvažuje a opakuje

Runtime vykoná nástroj

Flexibilnejší. Viac tokenov. **Ťažšie ladiť**, pretože chyby sa prenášajú naprieč krokmi. Agenti obaľujú LLM do reasoning loopu s nástrojmi na vykonávanie akcií.

Build with Claude Code: Nový Cohort

Spúšťame nový **2-dňový intenzívny kurz** postavený na cohortoch — *Build with Claude Code*, ktorý vyučuje John Kim, tréner stoviek inžinierov v Meta. Kurz začína **28. mája 2026**.

Agentic Loop

Context engineering a pamäťové vrstvy pre reálne projekty

MCPs & Hooks

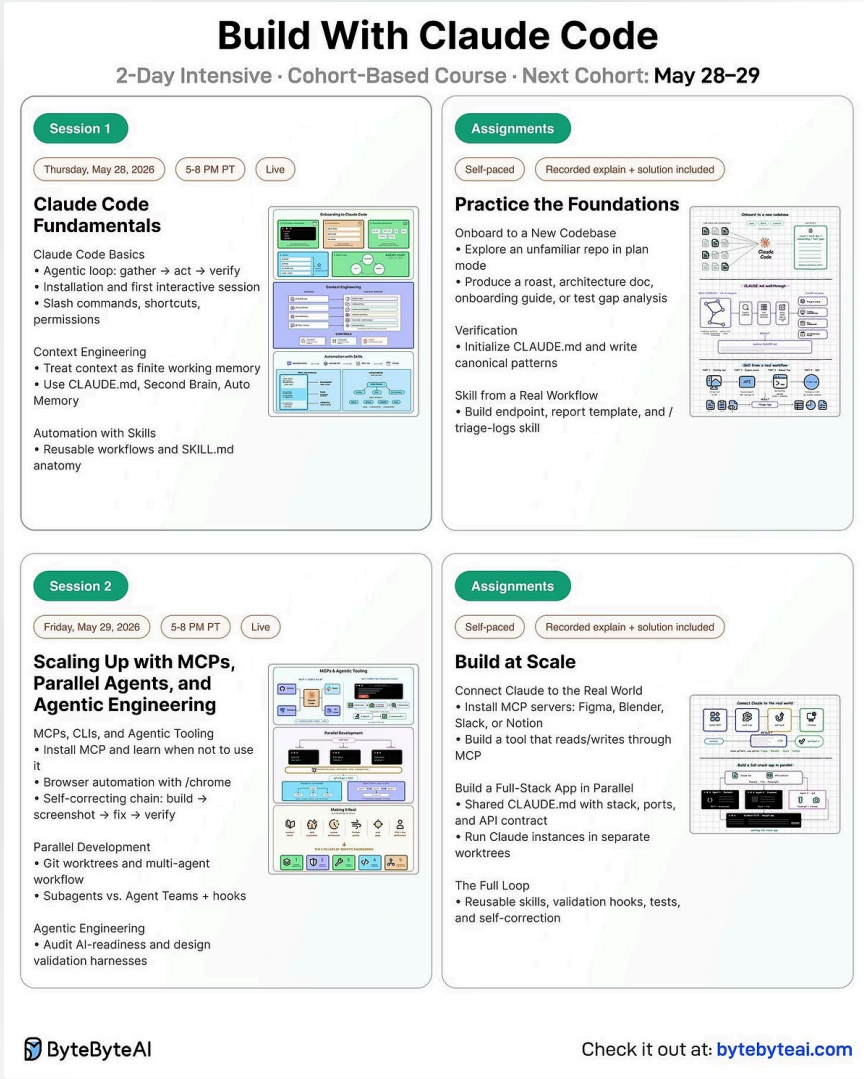
Nástroje a feedback looky pre sebakorekciu Claude Code

Paralelný vývoj

Git worktrees, subagenty a agent tímy

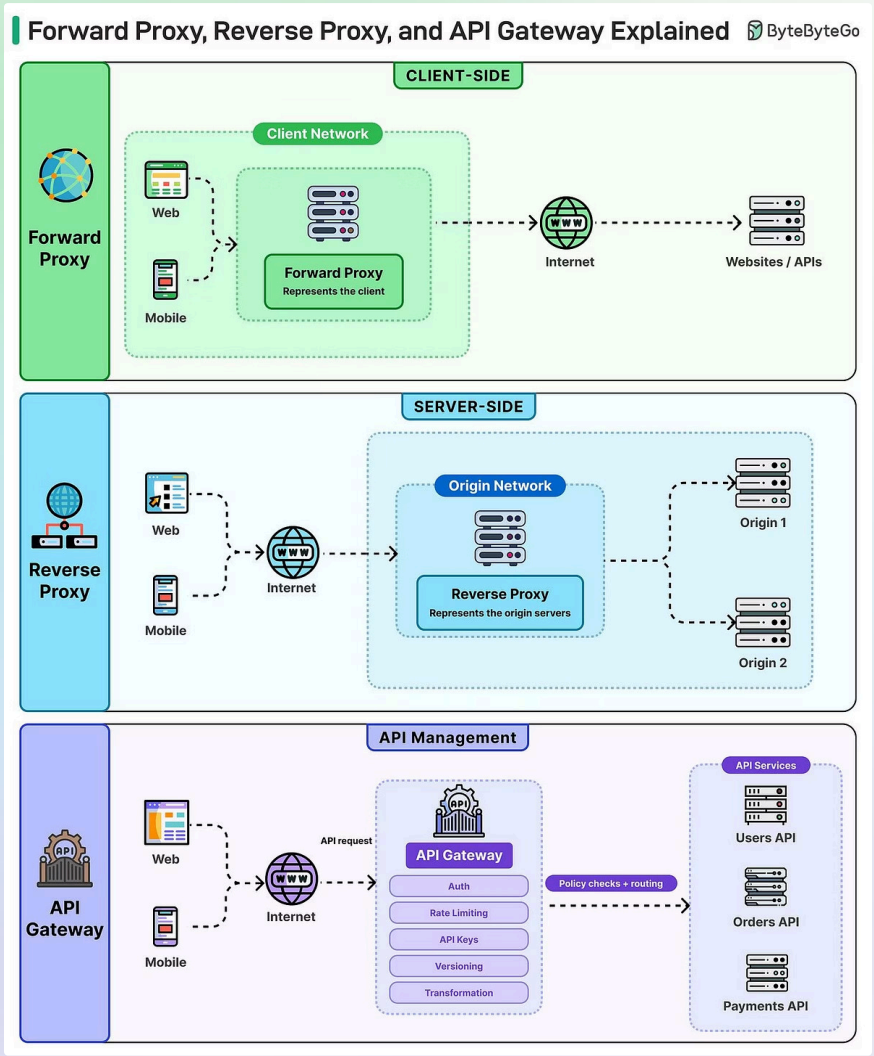
Capstone projekt

Odošlite niečo reálne na vlastnom stacku



Forward Proxy, Reverse Proxy a API Gateway

Ľudia ich často zamieňajú — všetky sedia medzi klientom a serverom. **Skutočný rozdiel je, ktorú stranu reprezentujú.**



Tri proxy vrstvy vysvetlené

Forward Proxy – strana klienta

Sedí vedľa klienta. Preposiela požiadavky von, cieľ nevidí vašu skutočnú IP. Firemné siete ho používajú na presadzovanie politík, blokovanie stránok a cachovanie prevádzky.

Reverse Proxy – strana servera

Sedí vedľa servera. Klient netuší, koľko strojov je za ním. Proxy rozhoduje, kto spracuje požiadavku, ukončuje TLS a drží backend mimo verejného internetu. Typicky NGINX alebo HAProxy.

API Gateway – viac než len routing

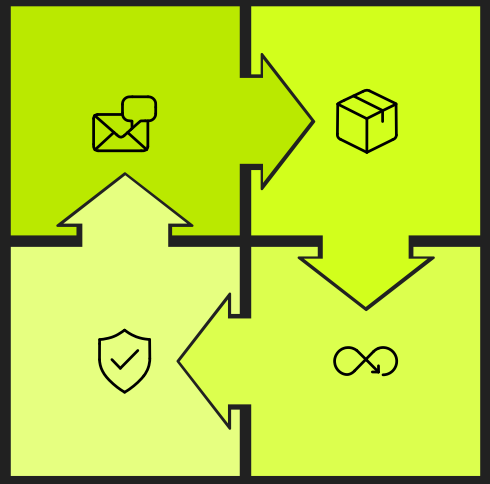
Reverse proxy, ktorý robí viac: auth, rate limity, API kľúče, verziovanie a tvarovanie požiadaviek. Bez neho musí každá mikroslužba implementovať vlastnú validáciu a throttling.

ⓘ V reálnych systémoch bežia všetky tri na rôznych vrstvách súčasne.

Ako požiadavka cestuje cez Claude Code

Sledujme reálnu požiadavku: *"Oprav zlyhávajúci test v auth.test.ts."* Celý systém je len tento loop, opakovaný, kým model neprestane žiadať nástroje.

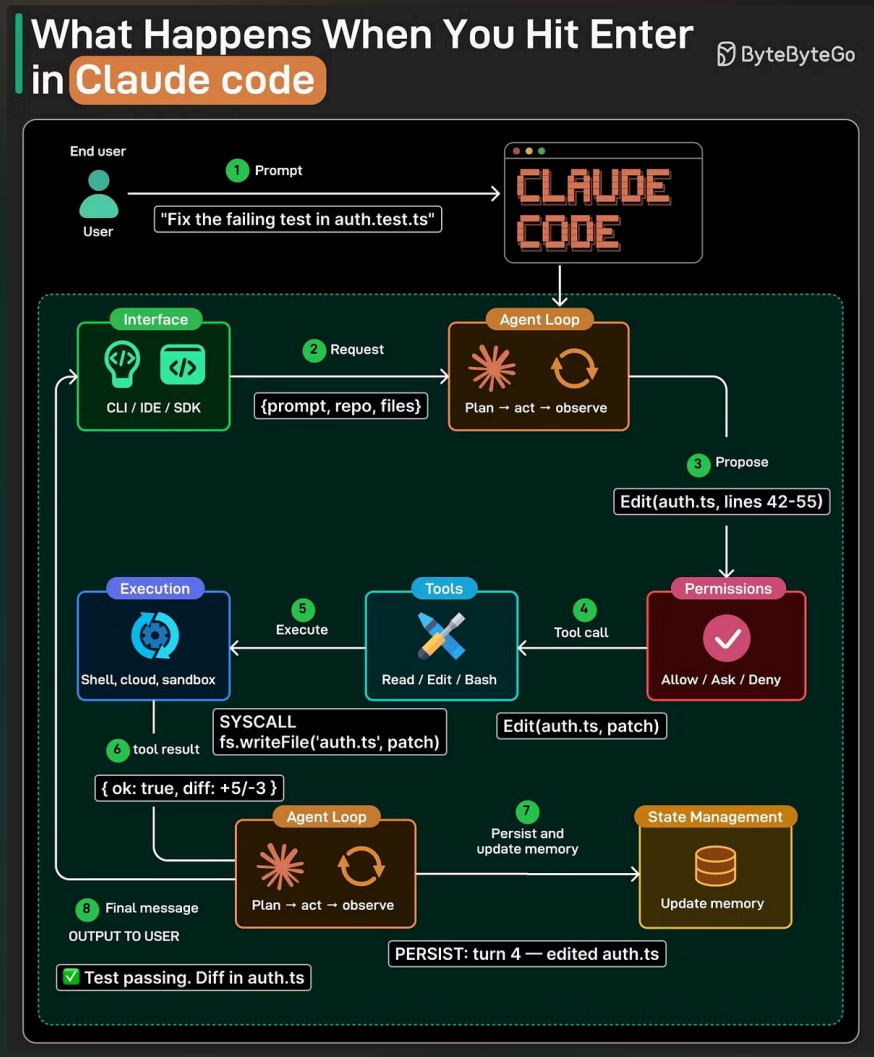
Prompt od používateľa



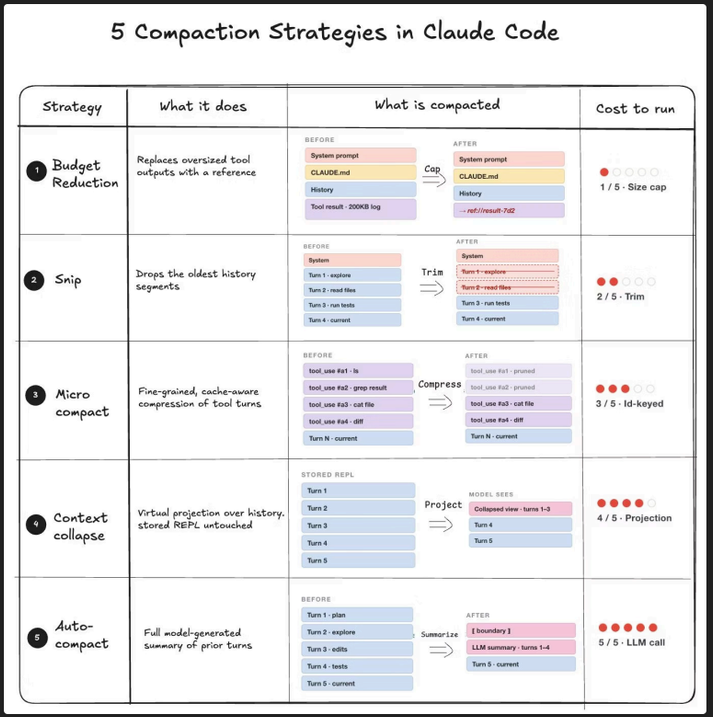
Interface zabalí kontext

Permission a tool call

Agent loop navrhne akciu



5 stratégií kompakcie kontextu v Claude Code



Claude Code používa **5 stratégií** spúšťaných v poradí pred každým volaním modelu. Každá sa spustí len ak predchádzajúca neuvoľní dostatok miesta — *lazy degradation*.

01

Budget Reduction

Obmedzí veľké výstupy nástrojov — nahradí ich odkazom na obsah.

02

Snip

Ostrihá najstaršie segmenty histórie a vloží hraničný marker.

03

Microcompact

Prune tool turns podľa tool_use_id, aby prompt cache zostala teplá.

04

Context Collapse

Read-time projekcia nad celou históriou — uložené REPLy sa nedotknú.

05

Auto-compact

Posledná záchrana: model vytvorí úplné zhrnutie predchádzajúcich turnov.

Kľúčové závery

RAG vs Agent

Použite RAG, keď odpoveď žije v dokumentoch. Použite agenta, keď úloha vyžaduje akcie na iných systémoch.

Proxy vrstvy

Forward proxy = klient. Reverse proxy = server. API Gateway = reverse proxy + auth, rate limiting a politiky.

Claude Code

Požiadavky prechádzajú 8-krokovým agent loopom. Kontext sa spravuje 5 stratégiami lazy degradácie.

Ďalšie čítanie: **Load Balancer vs API Gateway** a **Git Workflow: Essential Commands** — ByteByteGo newsletter.