

Systemový dizajn & QA – Prehľad pre inžinierov

Kompletný študijný materiál pokrývajúci kľúčové témy moderného softvérového inžinierstva: od AI-natívneho testovania cez RAG architektúry, Redis dátové štruktúry, bezpečnosť API, návrhové vzory až po testovací pyramídový model.

BYTEBYTEGO · SYSTÉM DESIGN REFRESHER



KAPITOLA 1

Prečo pomalé QA brzdí váš tím?

Pomalé QA procesy sú jedným z najväčších brzdiacich faktorov pri vydávaní softvéru. Manuálne end-to-end testovanie zaberá hodiny, chyby sa dostávajú do produkcie a vývojové tímy čakajú na výsledky testov namiesto toho, aby dodávali hodnotu.

QA Wolf je AI-natívna služba, ktorá podporuje webové aj mobilné aplikácie. Dosahuje **80 % automatizovaného pokrytia testami** v priebehu niekoľkých týždňov a pomáha tímom **dodávať 5x rýchlejšie** skrátením QA cyklov na minúty.

✗ Bez QA Wolf

- Manuálne E2E testovanie
- Pomalé QA cykly
- Chyby v produkcii

✓ S QA Wolf

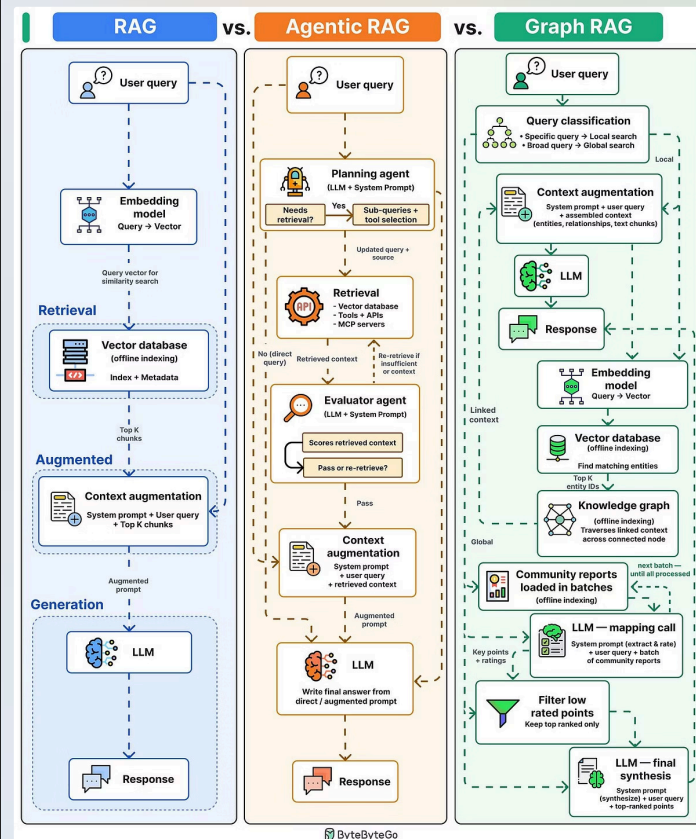
- Automatizované pokrytie
- QA cykly v minútach
- Zero flakes garancia



KAPITOLA 2

RAG vs. Graph RAG vs. Agentic RAG

RAG (Retrieval-Augmented Generation) prepája veľké jazykové modely s vašimi vlastnými dátami. Existujú tri hlavné prístupy – každý s inými silnými stránkami, nákladmi a vhodným použitím.



Porovnanie troch RAG prístupov



Štandardný RAG

Dotaz sa konvertuje na embedding a porovná s vektorovou databázou. Top-K najbližšie chunky sa odovzdajú LLM ako kontext a model vygeneruje odpoveď.

Vhodné pre: Rýchle odpovede z dokumentov, kde záleží na rýchlosti a cene.



Graph RAG

Dotaz je klasifikovaný: špecifické otázky idú cez lokálne vyhľadávanie (znalostný graf), všeobecné cez globálne (komunitné reporty v dávkach).

Vhodné pre: Štruktúrované znalosti – právne, biomedicínske alebo compliance dáta.



Agentic RAG

Reasoning agent rozloží dotaz na podotázky, vyberie zdroje, vytiahne kontext a evaluačný agent overí kvalitu odpovede. Pri nedostatočnom výsledku sa retrieval opakuje.

Vhodné pre: Viacstupňové uvažovanie a samo-korekciu pri komplexných dotazoch.

❏ Štandardný RAG = rýchly a lacný · Graph RAG = drahý na vybudovanie · Agentic RAG = najschopnejší, ale pomalší a ťažšie laditeľný

KAPITOLA 3

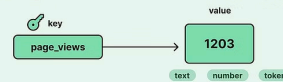
Redis dátové štruktúry pre každého inžiniera

Redis 8 obsahuje desať vstavaných dátových štruktúr. Každá má špecifický účel — správny výber zásadne ovplyvňuje výkon a jednoduchosť implementácie.

Redis Data Structures Every Engineer Should Know

ByteByteGo

String



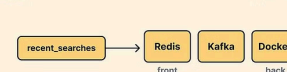
Best for: One value, counter, or token per key.

Hash



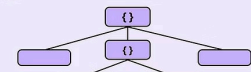
Best for: Storing an object's fields under one key.

List



Best for: Queues, stacks, and ordered sequences.

JSON



Best for: Nested documents you query and update in place.

Set



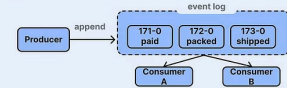
Best for: Unique members and overlap between groups.

Sorted Set



Best for: Leaderboards and ranking by score.

Stream



Best for: Event logs with multiple consumers.

Geospatial



Vector Set



Best for: Finding items most similar to a query.

Time Series



Best for: Metrics, telemetry, and timestamped measurements.

Redis – Základné (Core) štruktúry

String

Jedna hodnota na kľúč. Ideálne pre čítače, session tokeny a cachované payloady.

Hash

Polia objektu pod jedným kľúčom. Aktualizácia jedného poľa nevyžaduje prepis celého záznamu.

List

Usporiadané sekvencie s rýchlym push/pop na oboch koncoch. Fronty, feedy, zoznamy posledných položiek.

Set

Unikátni členovia s podporou prieniku, zjednotenia a rozdielu. Tagovanie, deduplikácia, follower overlap.

Sorted Set

Členovia zoradení podľa číselného skóre. Rebríčky, prioritné fronty, top-N dotazy.

Stream

Append-only log s consumer groups. Každý konzument sleduje vlastnú pozíciu, server sleduje nepotvrdené správy.

Redis – Špeciálne štruktúry

JSON

Vnorené dokumenty s JSONPath prístupom. Aktualizácia hlboko vnoreného poľa bez read-modify-write.

Geospatial

Indexy zemepisnej šírky/dĺžky s radius a box dotazmi. Interne implementované ako Sorted Set s geohash skóre.

Vector Set

Approximate nearest-neighbor vyhľadávanie nad embeddingmi. Základ retrieval kroku vo väčšine RAG pipeline.

Time Series

Timestampované vzorky s retenčnými politikami, downsamplingom a štítkami. Metriky, telemetria, IoT dáta.

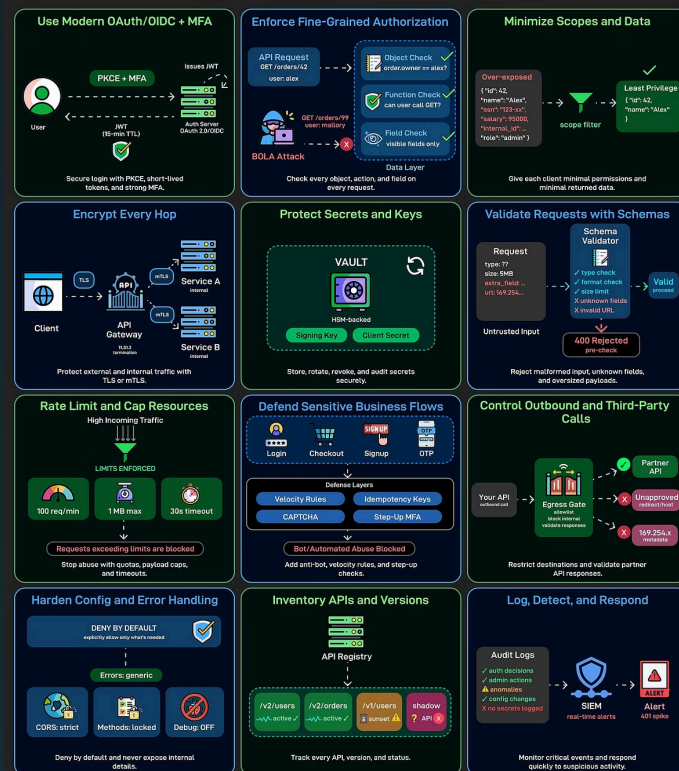
KAPITOLA 4

Bezpečnosť API – Najlepšie praktiky

Väčšina únikov cez API nastáva kvôli chybnjej autorizácii, úniku tajomstiev alebo chýbajúcim rate limitom. Toto sú základné princípy, ktoré by mal dodržiavať každý tím.

API Security Best Practices

ByteByteGo



12 pilierov bezpečnosti API



OAuth/OIDC + MFA

PKCE pre verejných klientov, krátkodobé tokeny a step-up MFA pre citlivé operácie. Implicit a password granty sú zastarané.



Jemno-zrnitá autorizácia

Kontrolujte oprávnenia na úrovni objektu, funkcie aj poľa pri každom requeste. BOLA je stále top zraniteľnosť API.



Ochrana tajomstiev

Ukladajte podpisové kľúče v HSM-zálohovaných vaultoch. Pravidelne ich rotujte a nikdy ich netlačte do repozitárov.



Šifrovanie každého skoku

TLS pre externý prenos, mTLS medzi službami. Ak prenos prechádza sieťovou hranicou, musí byť šifrovaný.



Rate limiting a obmedzenie zdrojov

Kvóty na používateľa, limity veľkosti payloadu a timeouty. Bez nich jeden zlý klient zhodí celý systém.

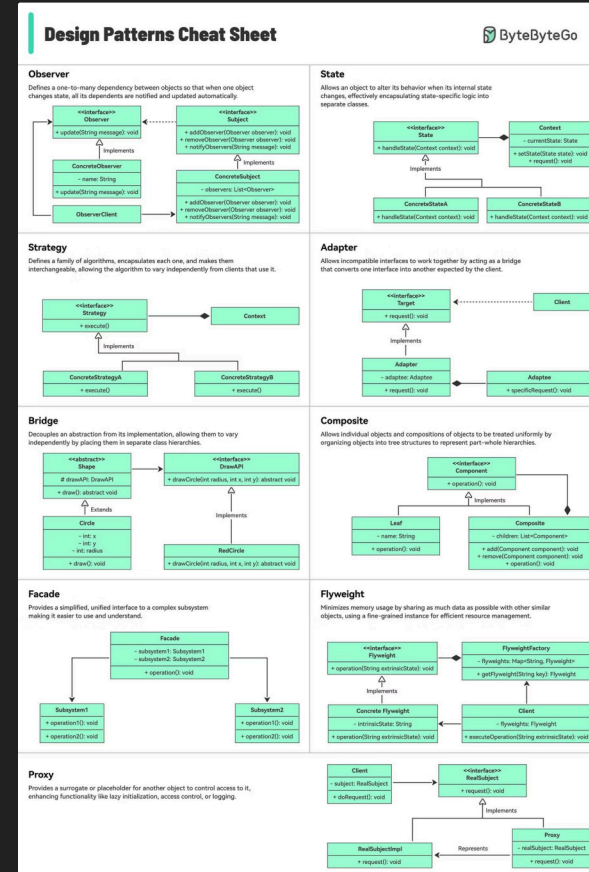
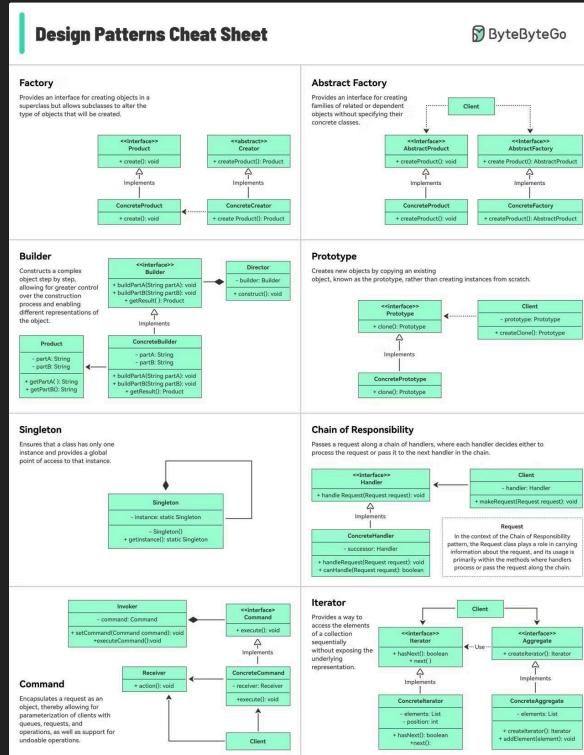


Logovanie a reakcia

Posielajte auth rozhodnutia a anomálie do SIEM. Upozornenie na skoky 401 pred tým, než sa stanú incidentmi.

Návrhové vzory – Cheat Sheet

Návrhové vzory sú overené riešenia opakujúcich sa problémov v softvérovom dizajne. Cheat sheet poskytuje stručné vysvetlenie každého vzoru a jeho použitie v praxi.



Kreačné vzory

Factory, Abstract Factory, Builder, Prototype, Singleton – riešia, ako sa objekty vytvárajú.

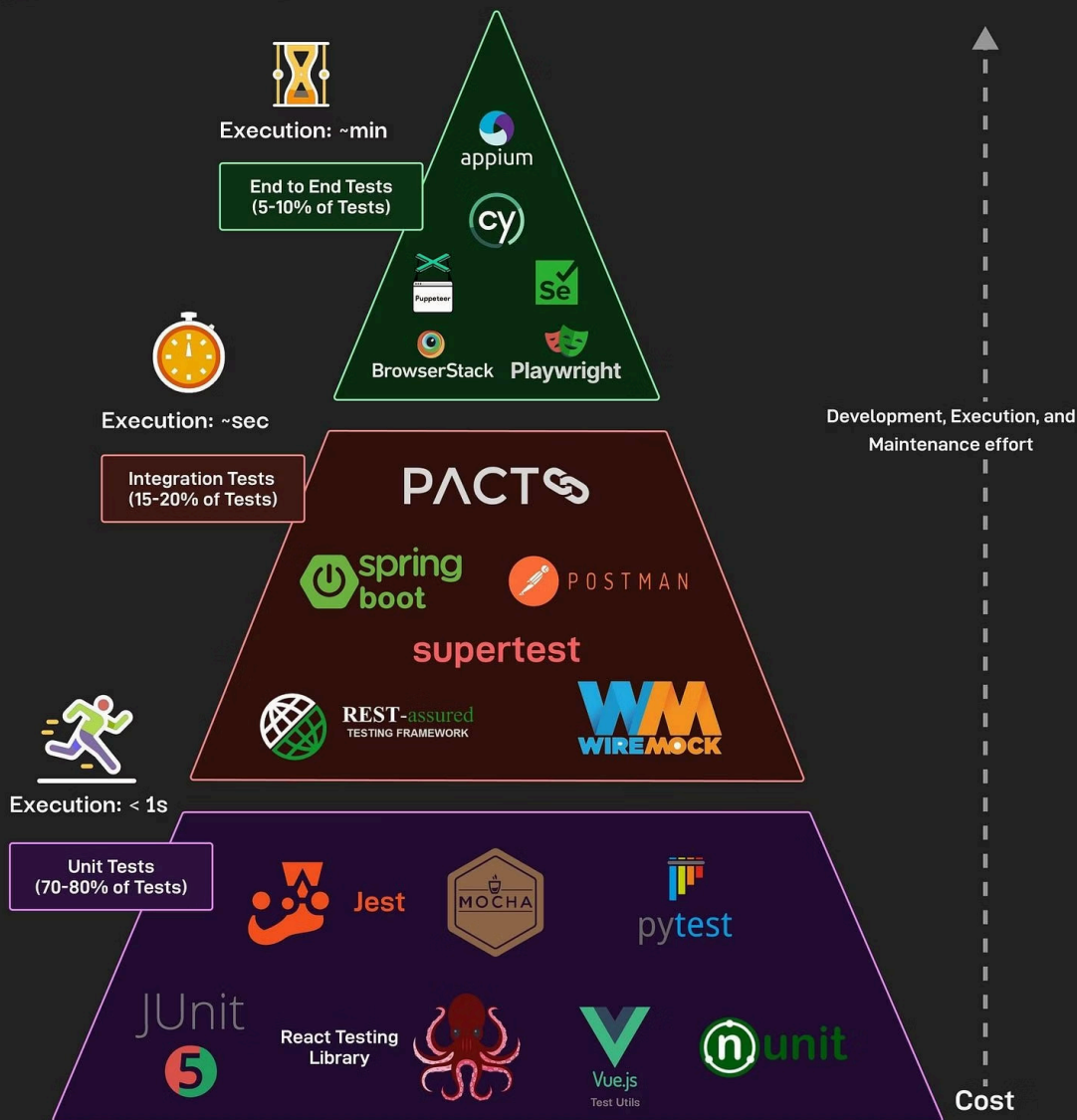
Štrukturálne vzory

Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy – riešia kompozíciu tried.

Behaviorálne vzory

Chain of Responsibility, Command, Iterator, Observer, State, Strategy – riešia komunikáciu medzi objektmi.

The Testing Pyramid



Čo sme prebrali dnes

01

QA Wolf

AI-natívne testovanie s 80 % pokrytím a 5× rýchlejšie vydávanie softvéru.

03

Redis Štruktúry

10 vstavaných štruktúr v Redis 8 — od Stringov a Hashov po Vector Set a Time Series.

05

Návrhové vzory

Kreačné, štrukturálne a behaviorálne vzory ako základ čistého softvérového dizajnu.

02

RAG Architektúry

Štandardný, Graph a Agentic RAG — každý pre iný typ problému a požiadavky.

04

Bezpečnosť API

12 pilierov: autorizácia, šifrovanie, rate limiting, logovanie a správa tajomstiev.

06

Testovací pyramídový model

Unit → Integrované → E2E: správny pomer testov pre rýchle, spoľahlivé a udržateľné vydávanie.