

Mikroservisy

Petr Paloch



Co jsou mikroservisy?

- Autonomně běžící služby
- Každá mikroservisa je samostatně běžící program
- Spolupracují za společným cílem
- Komunikují spolu specifikovaným způsobem
- Co nejmenší a zajišťují jen daný malý úkol
- Mohou být psané v různých programovacích jazycích
- Mohou používat různé databázové zdroje
- Mohou běžet na různých strojích s různými operačními systémy



Kdo používá mikroservisy?

- Comcast Cable
- Uber
- Netflix
- Amazon
- Ebay
- Sound Cloud
- Karma
- Groupon
- Hailo
- Gilt
- Zalando
- Capital One
- Lending Club
- AutoScout24

Nasazování a udržení v chodu

- automatizace sestavení a nasazování
- od deploye na GIT až po nasazení zautomatizovaně
- nejlépe více environmentů - vývojový a produkční
- mikroservisy je třeba udržovat běžící
- možnost spuštění ve více instancích



Kubernetes

- správa cloudu
- koordinuje cluster počítačů
- automatický deployment
- škálování a údržba
- cluster vystupuje jako jeden velký zdroj



Komunikace mikroservis

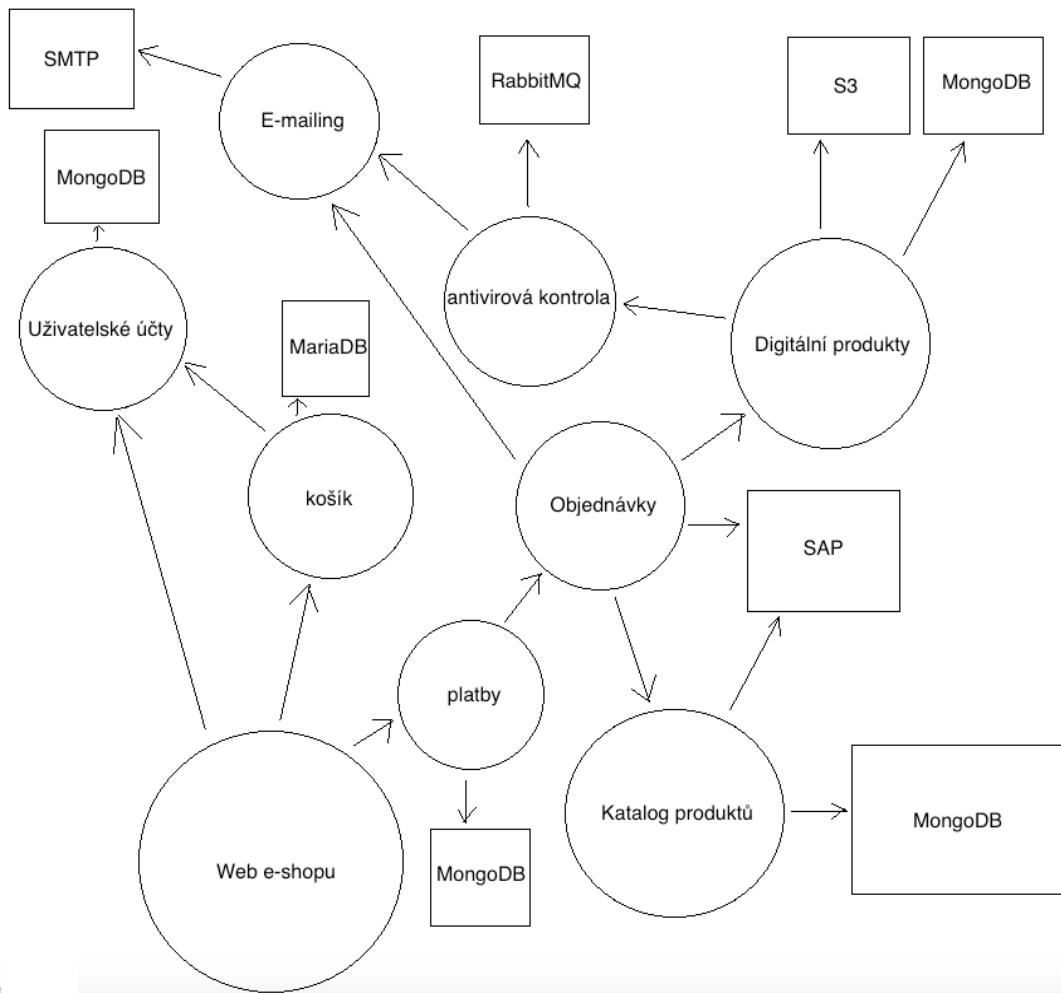
- API Brána - HTTP, JSON/XML
- možnost autentizace - kdo smí komunikovat s kým
- samostatná API - každá mikroservisa vystavuje URL
- API mají být interní
- řízení práv přístupu



Modelový příklad

e-shop

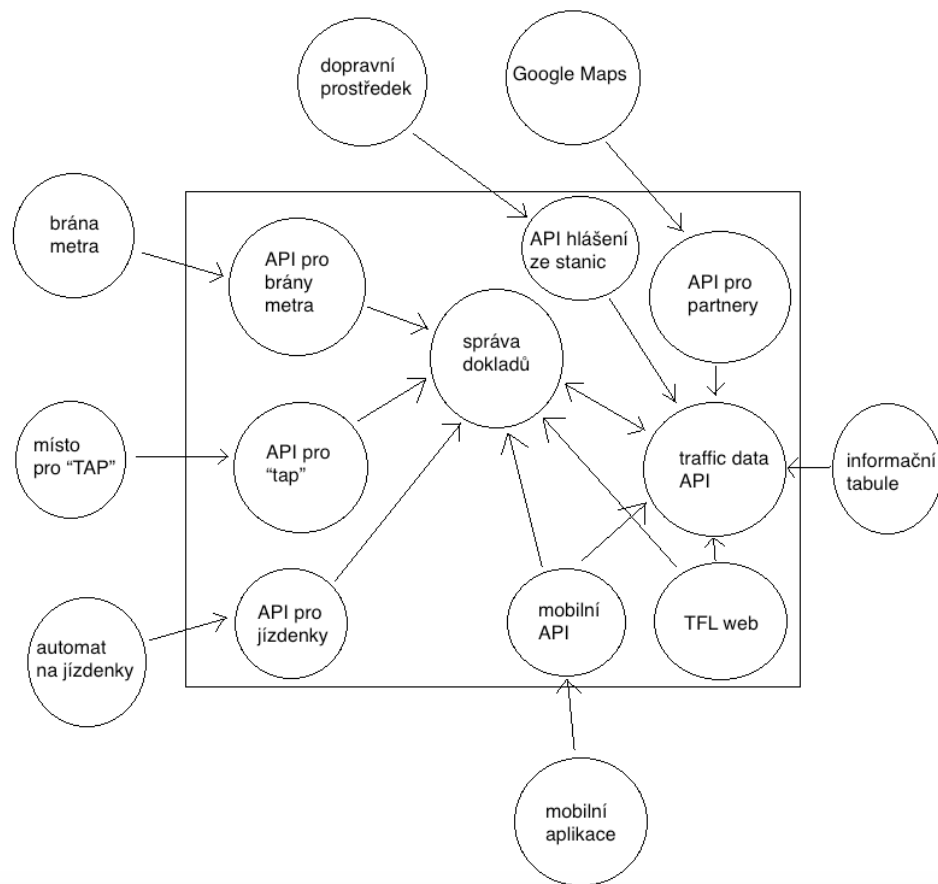
- prodává i digitální obsah
- digitální obsah odesílá na e-mail
- platební brána
- košík
- objednávky
- správa uživatelů
- katalog produktů v SAPu



Modelový příklad

Transport for London

- vstupní brány metra
- “tap” systémy v autobusech a některých zastávkách
- kiosky na jízdenky
- informační tabule ve stanicích
- mobilní aplikace uživatelů
- TFL web - “I checked my road”
- hlášení polohy ve stanicích
- API pro partnery



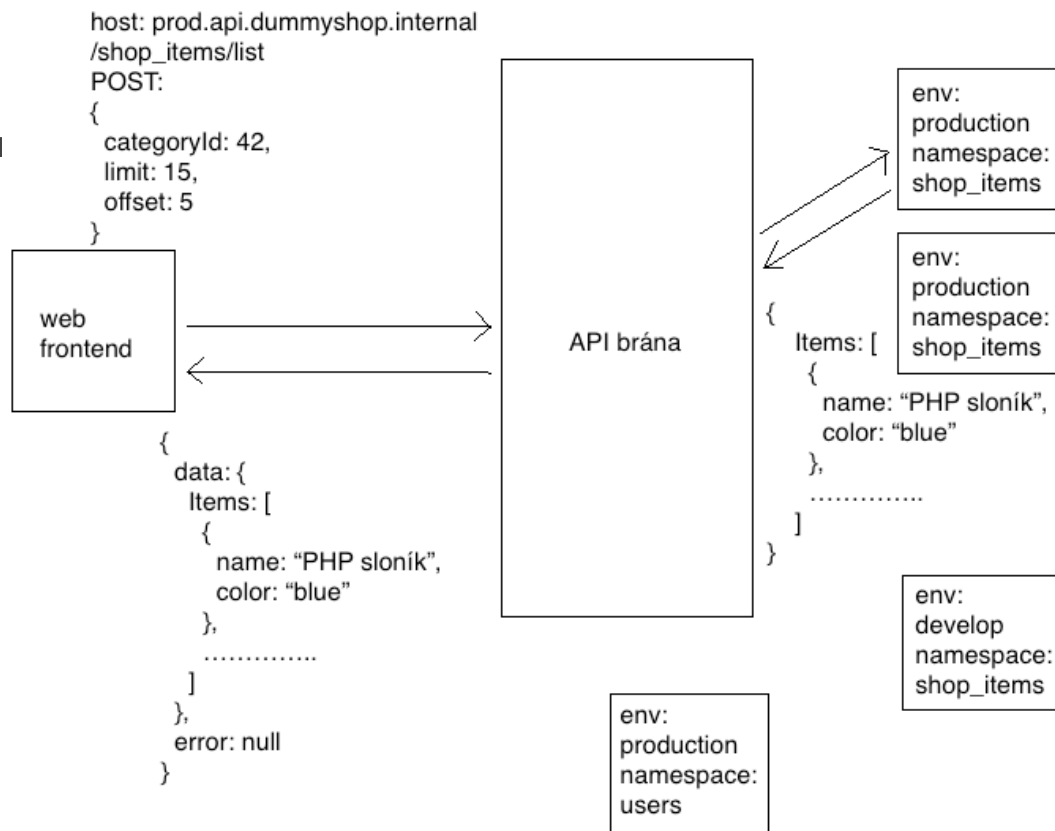
API brána

- Jednotlivé mikroservisy jsou připojeny na bránu
- Nekomunikují přímo spolu
- Možnost rozdělit provoz na více instancí dané mikroservisy
- Brána může řídit oprávnění - kdo komunikuje s kým
- Může přidělit dané mikroservise konkrétní API URL adresy
- Může umožnit běh pro více environmentů - například produkční a vývojový



API brána

- více prostředí rozlišeno doménou
- žádáme o list produktů v kategorii
- více mikroservis v namespace
- load balancing
- brána předává data z mikroservisů



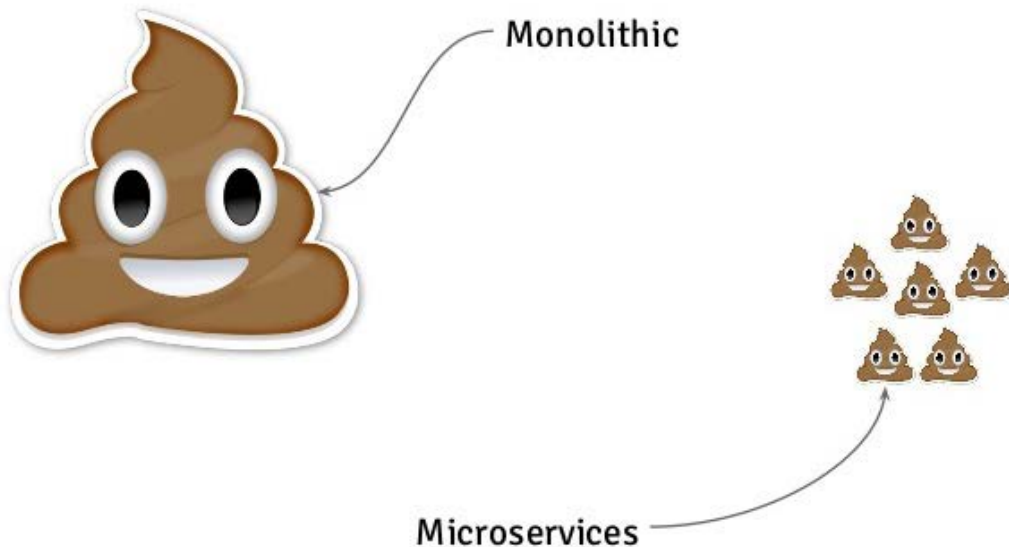
Výhody mikroservis

- izoluje chyby - spadne nám daná mikroservisa, nikoliv celý monolit
- skrývá implementační detaily
- deployujeme nezávisle - rozdělení na týmy
- decentralizuje jednotlivé úkoly
- každá microservica je vlastní projekt na vlastním repozitáři
- pokud se na určité úkoly hodí určitý programovací jazyk - použijme ho
- pokud se hodí jiný databázový systém pro určitý úkol - použijeme ho :)
- dobrá znalost svojí části kódu



Nevýhody mikroservis

Monolithic vs Microservices



Nevýhody mikroservis

- Složitější na údržbu
- Vyšší provozní režie
- Vyšší síťová režie - komunikace mikroservis
- Nutnost údržby rozhraní mikroservis
- Složitější provádění transakcí v databázi



Monolit

- celá aplikace psaná jako jeden celek
- většinou používá jeden databázový zdroj
- běží na jednom stroji



Výhody monolitu

- Není nutné udržovat architekturu pro běh mikroservis
- Aplikace je jeden celek
- Jeden projekt na GITu
- Jednodušší řešení sestavování a nasazování
- Jednodušší přenositelnost na jiný server/hosting/cloud



Přechod na mikroservisy

- Nejlepší provádět po menších částech
- Postupné rozsekávání monolitu
- Tvorba mikroservis dle daného účelu
- Napojování nových mikroservis na monolit
- Nahrazování společných částí v různých projektech



Dotazy?

?



Díky za pozornost!

