



Spis treści

1	Wstęp	2
1.1	Wprowadzenie	2

1 Wstęp

1.1 Wprowadzenie

W ostatnich latach w świecie IT można było zauważyć trend polegający na promowaniu budowania dużych systemów jako aplikacji rozproszonych składających się z wielu serwisów. W dniu dzisiejszym rozwiązanie to określane jest architekturą **mikroserwisową**. Sama koncepcja nie jest czymś zupełnie nowym i możemy określić ją jako rozwinięcie architektury SOA (Service Oriented Architecture). Idea tworzenia oprogramowania w architekturze mikroserwisowej polega na budowaniu niewielkich autonomicznych komponentów z których każdy odpowiada za konkretne zadanie. Elementy te współpracując ściśle ze sobą pozwalają na dostarczenie wymaganej logiki biznesowej.

Wielu architektów oraz zespołów programistów wizja rozbicia swojej monolitycznej aplikacji na architekturę mikroserwisową zachęciła do prób budowania takich rozwiązań. Pomimo początkowego entuzjazmu popartego niezaprzeczalnymi zaletami mikroserwisów takimi jak:

- możliwości stosowania różnych technologii
- skalowalności
- odporności na awarie

dały o sobie znać wady, które spowodowały, że projektowanie, implementacja oraz utrzymanie takiej architektury stało się olbrzymim wyzwaniem dla firm produkujących oprogramowanie. Jednym z największych problemów okazał się sposób komunikacji pomiędzy kolejnymi mikroserwisami. Budowa **API** wymagała od architektów rozwiązania problemów w następujących kwestiach:

- wyboru formatu wymiany danych (JSON, XML itp.)
- zaprojektowania endpointów

- obsługi błędów
- wydajności (ilość danych przy jednym wywołaniu serwisu, czas oczekiwania na odpowiedź)