

1.2.1. Wyzwania związane z projektowaniem

W jaki sposób te wyzwania wpływają na fazę projektowania i uruchamiania mikroserwisów? Wcześniej wprowadziliśmy pięć podstawowych zasad tworzenia mikroserwisów. Pierwszą z nich była *autonomia*. Aby nasze usługi były autonomiczne, musimy je tak zaprojektować, aby razem były luźno powiązane i każda zawierała wysoce spójne elementy funkcjonalności. Jest to proces ewolucyjny. Zakres naszych usług może się zmieniać w miarę upływu czasu, a my będziemy często się decydować na wyodrębnienie nowej funkcjonalności z usług, a nawet wycofanie już istniejących.

Dokonywanie tych wyborów jest wyzwaniem, szczególnie istotnym na początku opracowywania aplikacji! Głównym źródłem luźnych powiązań są granice między usługami; zły wybór doprowadzi do usług, które są trudne do zmiany i, ogólnie, mniej plastyczne i elastyczne do zastosowania.

OKREŚLANIE ZAKRESU MIKROSERWISÓW WYMAGA WIEDZY NA TEMAT DOMENY ICH ZASTOSOWANIA

Każdy mikroserwis jest odpowiedzialny za pojedynczą zdolność. Identyfikacja tych zdolności wymaga znajomości domeny biznesowej aplikacji. W początkowej fazie życia aplikacji nasza wiedza o domenie może być w najlepszym razie niekompletna lub, w najgorszym, niepoprawna.

Nieodpowiednie zrozumienie domeny problemu może skutkować złymi wyborami projektowymi. W aplikacji mikroserwisowej zwiększona sztywność granicy usługi w porównaniu z modułem w aplikacji monolitycznej oznacza, że koszt złych decyzji dotyczących zakresu może być wyższy:

- być może trzeba będzie refaktoryzować wiele różnych baz kodów;
- konieczna może być migracja danych z bazy danych jednej usługi do drugiej;
- możliwe, że nie zostaną zidentyfikowane niejawne zależności między usługami, co może prowadzić do błędów lub niezgodności podczas wdrażania.

Czynności te zilustrowano na rysunku 1.5.

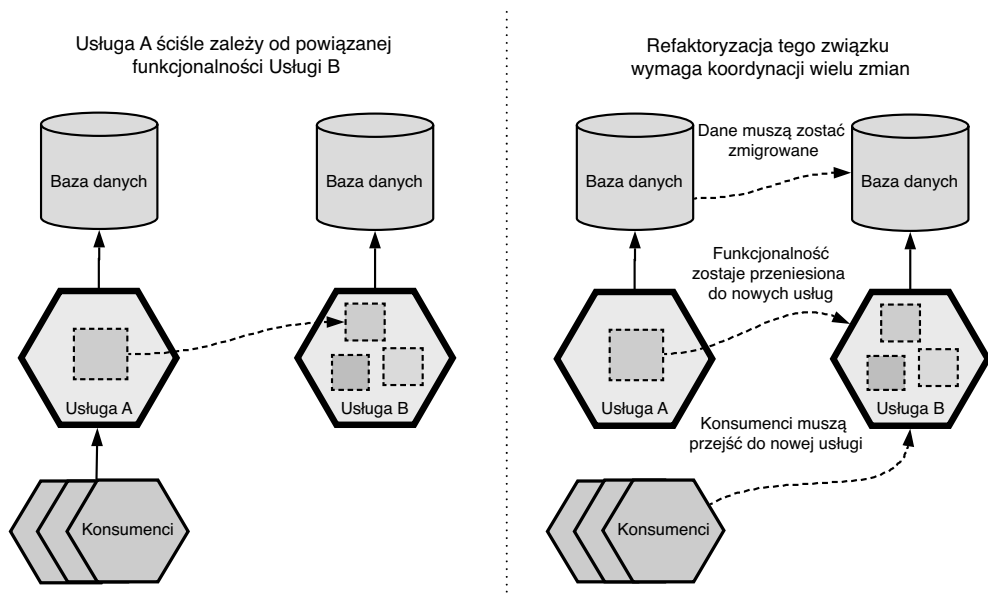
Jednak podejmowanie decyzji projektowych, przy niewystarczającej wiedzy o danej dziedzinie, nie jest rzadkością w przypadku mikroserwisów! Różne są skutki tych decyzji.

UWAGA W rozdziałach 2 i 4 za pomocą przykładowej aplikacji omówimy dobre praktyki dotyczące identyfikowania i określania zakresów usług.

ZARZĄDZANIE KONTRAKTAMI MIĘDZY USŁUGAMI

Każdy mikroserwis powinien być niezależny od sposobu implementacji innych usług. Pozwala to na techniczną heterogeniczność i autonomię. Aby to działało, każdy mikroserwis powinien wystawiać *kontrakt* – analogicznie do interfejsu w projektowaniu obiektowym – określający komunikaty, które spodziewa się otrzymywać i na które będzie udzielać odpowiedzi. Dobry kontrakt powinien być:

- *kompletny* – definiuje pełny zakres interakcji;
- *zwięzły* – nie wymaga więcej informacji niż jest to konieczne, tak, aby konsumenci mogli konstruować komunikaty w rozsądnych granicach;
- *przewidywalny* – dokładnie odzwierciedla rzeczywiste zachowanie implementacji.



Rysunek 1.5. Nieprawidłowe decyzje dotyczące zakresu usług mogą wymagać złożonej i kosztownej refaktoryzacji w obrębie granic usług

Każdy kto projektował API, wie, jak trudno osiągnąć takie cechy. Kontrakty są spoiwem między usługami. Z czasem kontrakty mogą ewoluować, ale jednocześnie muszą zachować wsteczną kompatybilność dla istniejących współpracowników. Te sprzeczne dążenia – między stabilnością a zmianą – są trudne do zarządzania.

APLIKACJE MIKROSERWISOWE SĄ PROJEKTOWANE PRZEZ ZESPOŁY

W większych firmach prawdopodobnie wiele zespołów będzie budować i uruchamiać aplikację mikroserwisową, z których każdy będzie odpowiedzialny za różne mikroserwisy. Każdy zespół może mieć własne cele, sposób pracy i cykl życia dostarczania. Projektowanie spójnego systemu może być trudne, gdy trzeba pogodzić harmonogramy i priorytety wielu niezależnych zespołów. Koordynacja rozwoju poważniejszych aplikacji mikroserwisowych będzie zatem wymagać porozumienia i uzgodnienia priorytetów oraz praktyk w wielu zespołach.

APLIKACJE MIKROSERWISOWE SĄ SYSTEMAMI ROZPROSZONYMI

Projektowanie aplikacji mikroserwisowych oznacza projektowanie systemów rozproszonych. W projektowaniu systemów rozproszonych pojawia się wiele błędnych założeń, takich jak:

- sieć jest niezawodna;
- opóźnienie jest zerowe;
- przepustowość jest nieograniczona;
- koszt transportu jest zerowy.