Pod lupę wziąłem trochę więcej wzorców niż w poleceniu, ale dużo jest według mnie godnych zastosowania.

**THROTTLING**

Ograniczenie ilości zapytań podczas krótkotrwałych okresów dużego popytu.

Dodatkowo dzięki throttlingowi jesteśmy w stanie zagwarantować dostęp do usługi zgodny z SLA bez konieczności auto skalowania.

Zaletą throttlingu jest informacja zwrotna którą otrzymuje aplikacja kliencka czy klient o przekroczeniu ilości zapytań.

Przykład usługi do zastosowania: API Management

**AUTOSCALLING:**

Zdolność aplikacji do dynamicznej zmiany swoich rozmiarów, dzięki czemu uzyskujemy wysoką wydajność i dostępność nawet w krytycznych momentach takich jak np. black Friday.

Jeżeli większość naszych klientów jest np. z Europy to możemy zmniejszyć liczbę instancji w godzinach mniejszego zapotrzebowania np. w nocy (po wcześniejszej dokładnej analizie ruchu).

Przykład usługi do zastosowania: tutaj raczej opcje wbudowane w usługi wystarczą czy to VM, VM Scale Sety czy skalowanie AppService. Należy tylko pamiętać o restarcie VM podczas resize (jeżeli byłaby potrzeba na skalowanie w górę a nie wszerz).

**Static Content Hosting:**

Aplikacje e-commerce np. sklepy internetowe posiadają bardzo dużą ilość statycznej treści chociażby obrazy.

Zwiększenie wydajności aplikacji poprzez wzrost szybkości dostępu do danej treści oraz nie obciążanie aplikacji.

Zmniejszenie kosztów magazynowania.

Przykład usługi do zastosowania: Content Delivery Network

**Sharding**

Tutaj od razu nasuwa mi się jako pierwsze RODO i inne kwestie związane z bezpieczeństwem.

Niektóre regiony prawnie wymagają aby dane ich mieszkańców mogły być przechowywane tylko w danej lokalizacji.

Dodatkowo w przypadku pojedynczego miejsca składowania i dużej ilości zapytań / operacji wykonywanych na danych może spaść wydajność i czas odpowiedzi.

Ograniczenia i limity wynikające z przechowywania danych w jednym miejscu. Systemy e-commerce przechowują bardzo dużą ilość danych, aby nie doprowadzić do sytuacji osiągnięcia limitu warto te dane dzielić.

Przykład usługi do zastosowania: Azure SQL Database, Azure CosmosDb

**Queue Based Load Leveling**

Dzięki użyciu kolejki usługa konsumuje w swoim tempie przychodzące wiadomości, nie wystąpi sytuacja zgubionej wiadomości nawet w przypadku dużego obciążenia aplikacji.

Jeżeli aplikacja jest typu „loosely coupled” zastosowanie kolejek nie wpłynie na dostępność aplikacji gdy jeden z konsumentów nie jest dostępny to i tak inna usługa może wrzucać swoje dane do kolejki nie powodując przestoju w pracy aplikacji.

Przykład usługi do zastosowania: Azure Service Bus

Zalety: możliwość tworzenia tematów, które w przeciwieństwie do zwykłej kolejki nie konkurują o dane.

**Load Balancing**

Nasza usługa musi być wysoce dostępna i wydajna. Zatem przychodzący ruch należy rozdzielać.

Usługa niezbędna do przeprowdzania konserwacji aplikacji bez przestojów w jej działaniu.

Przykład usługi do zastosowania: Load Balancer, Traffic Manager

Jeżeli np. nasi klienci są z całego świata warto zastosować Traffic Manager, który ma opcję przekierowywania klienta do najbliższego punktu końcowego tak aby opóźnienie było jak najmniejsze. Zastosowanie w przypadku App Service'ów.

W przypadku aplikacji opartej na wirtualnych maszynach zastosowałbym Load Balancer.

Jeżeli potrzebne byłoby rozrzucanie ruchu na poziomie warstwy 7 to można zastosować jeszcze Application Gataway choć to dość kosztowne rozwiązanie.