Cloud Design Patterns

Competing Consumers
Queue-based Levelling
Throttling

Co mají společného?

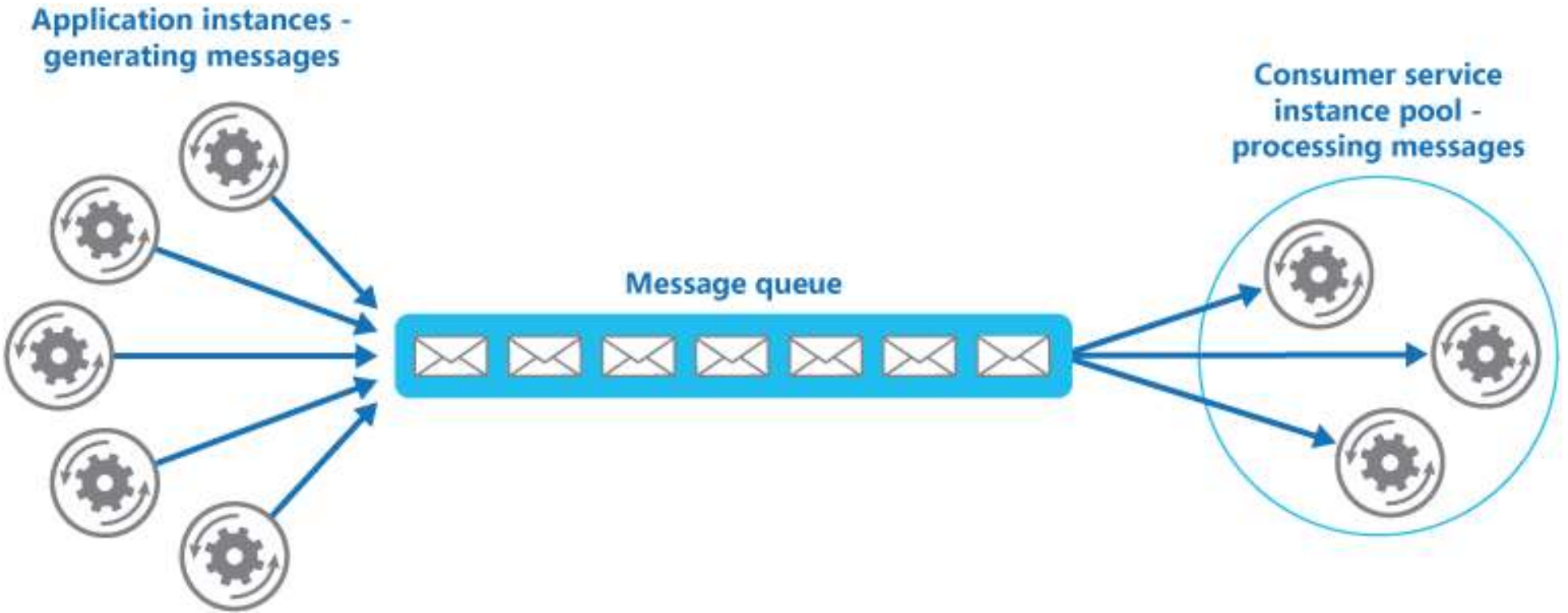
- Cloud patterny
- Přicházejí nám náhodně zprávy
- Chceme zabezpečit dostupnost

Competing Consumers

- „Umožňuje několika souběžným příjemcům zpracovávat zprávy přijaté ve stejném kanálu pro zasílání zpráv.“

https://docs.microsoft.com/cs-cz/azure/architecture/patterns/competing-consumers

Competing Consumers



Competing Consumers - Výhody

- Systém vyrovnání
- Zvyšuje spolehlivost
- Nevyžaduje složitou koordinaci
- Škálovatelnost

Competing Consumers - Problémy

- Pořadí zpráv
- Odolnost
- Poškozené zprávy
- Více front?
- Zajištění spolehlivosti

Kdy **ANO**

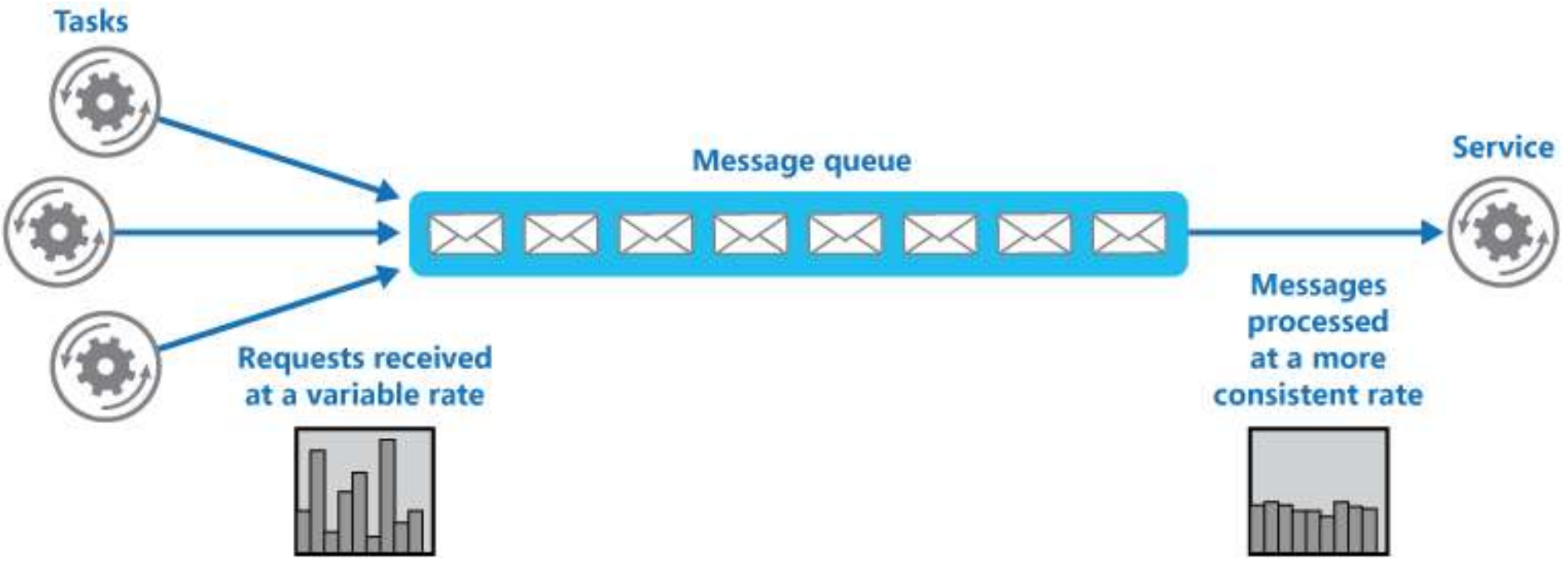
- Rozdělitelná logika
- Paralizovatelnost
- Proměnlivé množství úloh
- Vysoká dostupnost

Queue-Based Load Leveling

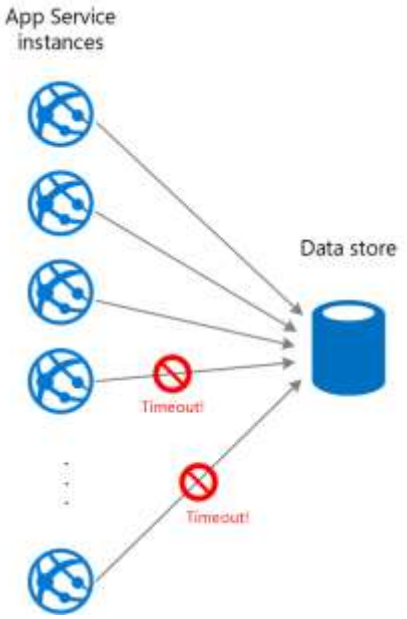
- „Použije frontu, která funguje jako vyrovnávací paměť mezi úlohou a službou, kterou vyvolá, pro ulehčení občasných velkých zátěží.“

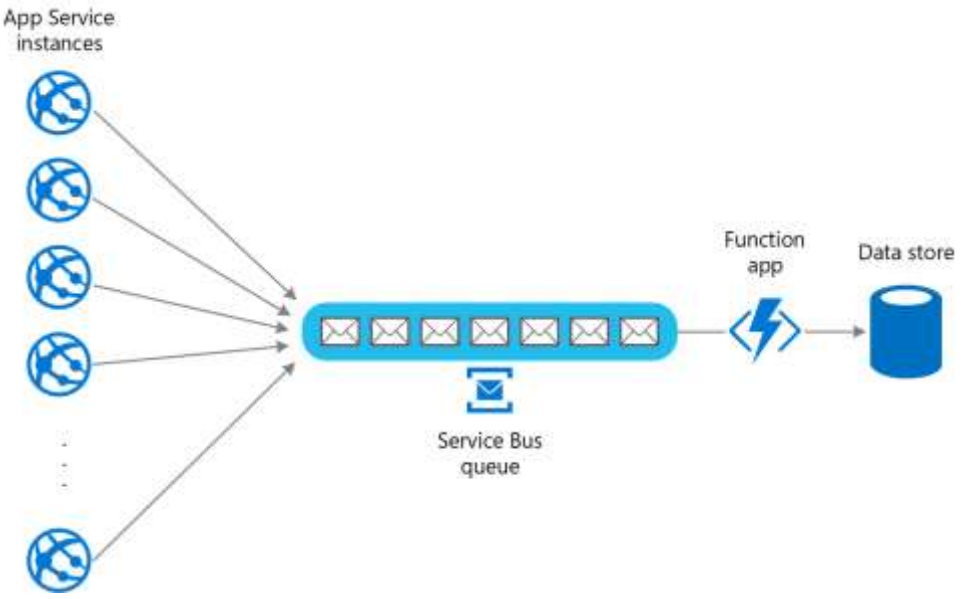
https://docs.microsoft.com/cs-cz/azure/architecture/patterns/queue-based-load-leveling

Queue-Based Load Leveling



Queue-Based Load Leveling





Queue-Based Load Leveling - Výhody

- ±Maximalizace dostupnosti
- ±Maximalizace škálovatelnosti
- Snížení nákladů

Queue-Based Load Leveling - Problémy

- Je potřeba logika okolo rychlosti zpracování
- Jednosměrnost fronty

Kdy **ANO**

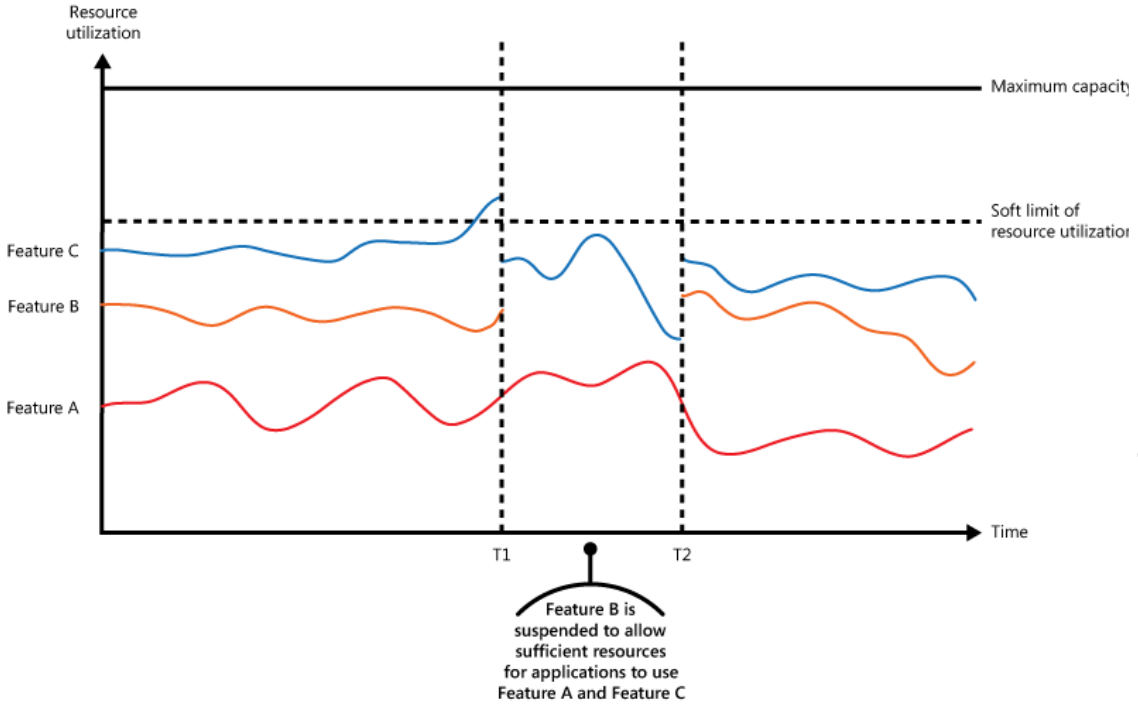
- Služby, které podléhají přetížení
- Není potřeba minimální latence

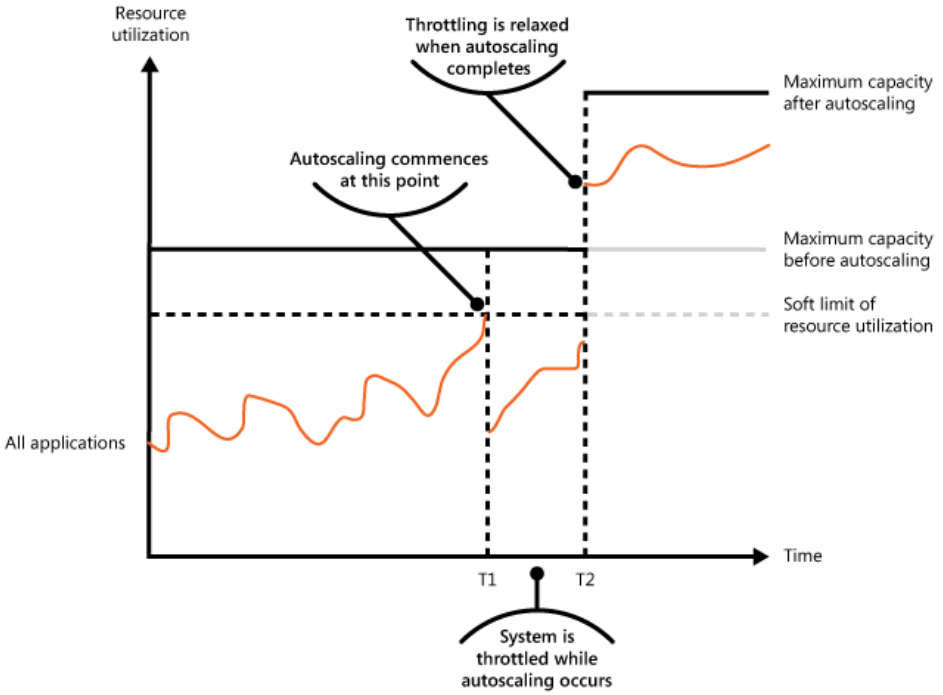
Throttling

- „Řídí spotřebu prostředků používaných instancí aplikace, jednotlivým tenantem nebo celou službou.“

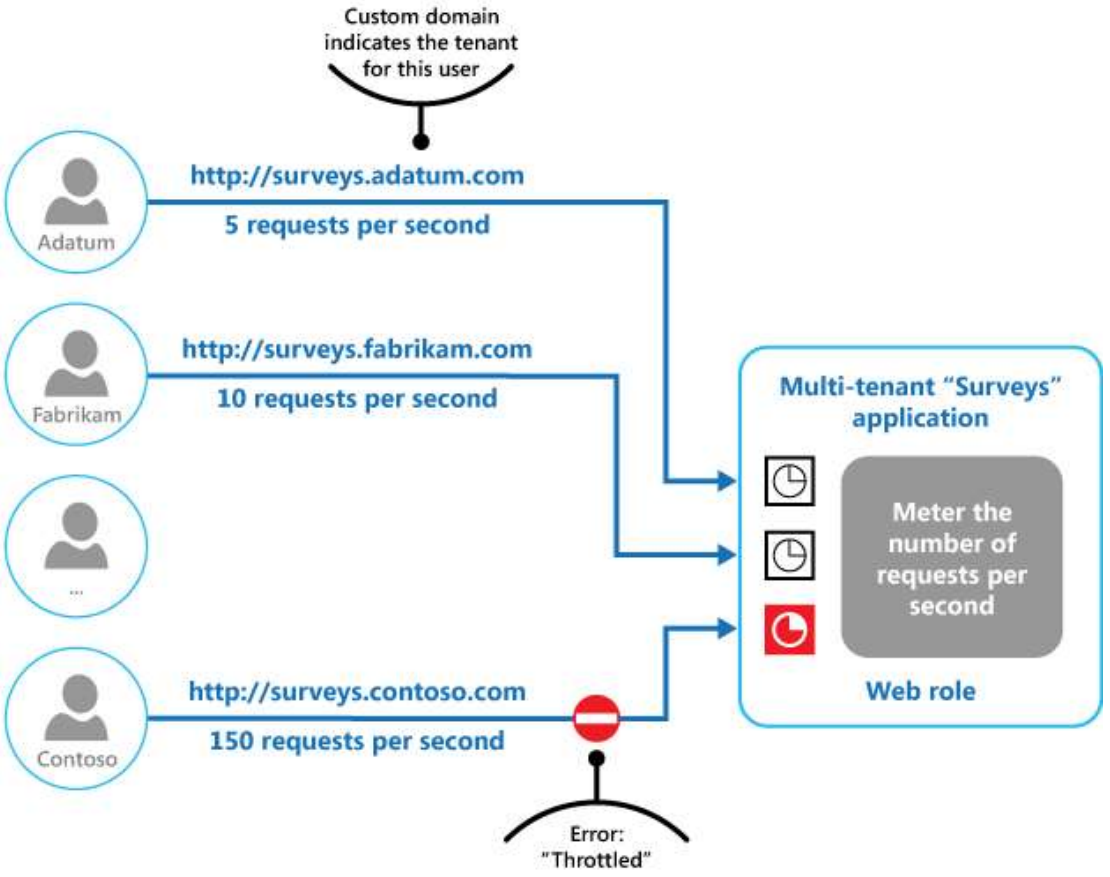
https://docs.microsoft.com/cs-cz/azure/architecture/patterns/throttling

Throttling


Resource utilization
Maximum capacity
Soft limit of resource utilization
Feature C
Feature B
Feature A
Time
T1
T2
Feature B is suspended to allow sufficient resources for applications to use Feature A and Feature C


Resource utilization
Throttling is relaxed when autoscaling completes
Maximum capacity after autoscaling
Autoscaling commences at this point
Maximum capacity before autoscaling
Soft limit of resource utilization
All applications
Time
T1
T2
System is throttled while autoscaling occurs

Throttling


Custom domain indicates the tenant for this user
Adatum
http://surveys.adatum.com
5 requests per second
Fabrikam
http://surveys.fabrikam.com
10 requests per second
...
Contoso
http://surveys.contoso.com
150 requests per second
Error: "Throttled"
Multi-tenant "Surveys" application
Meter the number of requests per second
Web role

Throttling - Výhody

- Výhodnější než škálování na krátkou dobu

Throttling - Problémy

- Je nutné plánovat od začátku
- Omezení musí být rychlé
- Návratový kód v případě chyby
- Riziko nekorektního chování po čas škálování

Kdy **ANO**

- Je potřeba zajištění podmnožiny služeb
- Nikdo nám nesmí sebrat prostředky
- Vysoké nárůsty aktivity
- Optimalizace nákladů

Cloud Design Patterns
- Competing Consumers
- Queue-based Levelling
- Throttling

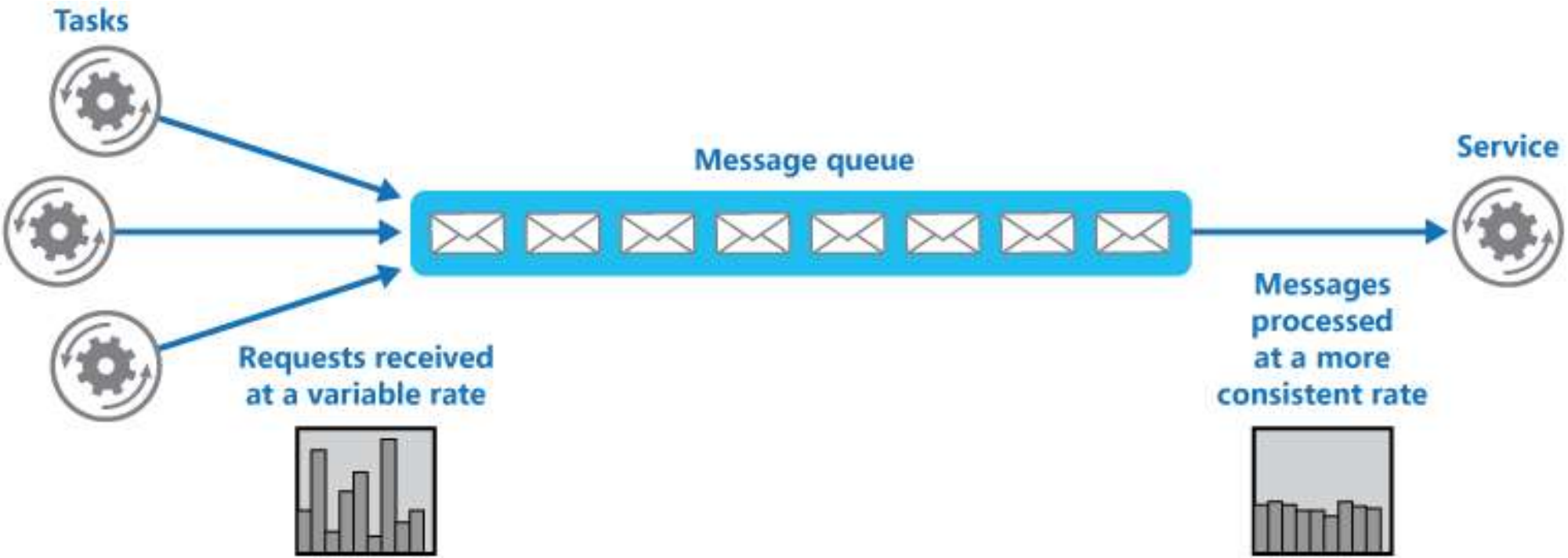
Děkuji za pozornost

Q&A

Queue-Based Load Leveling

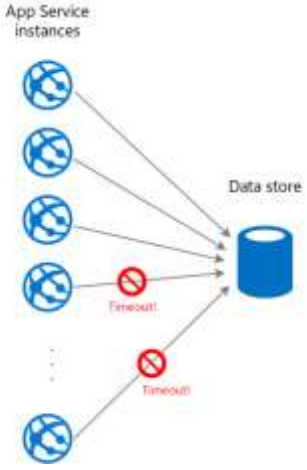
Q1: Požadavky z fronty si odebírá ten kdo je zpracovává, nebo jsou pouze preposílány?

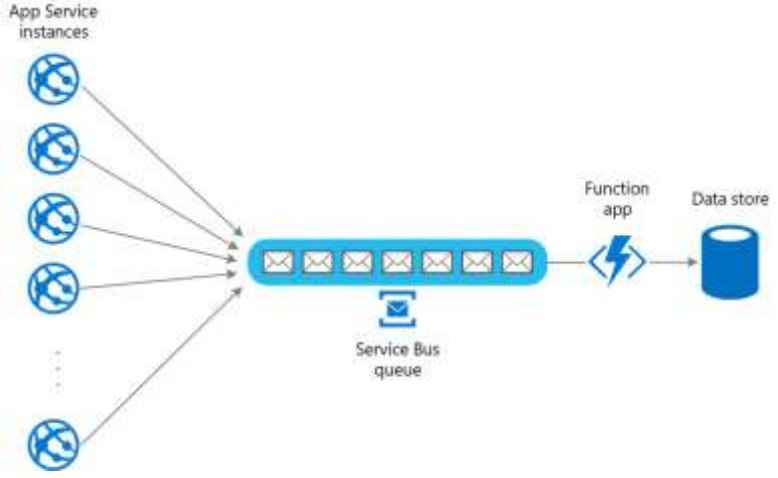
Q2: Požadavky se z fronty dostávají hned jak je možné, nebo dává smysl v nejakém prípade požadavky



Competing Consumers, Queue-based Leveling, Throttling

Q3: Bolo by možné ukázat príklady aspon v nejakom jednoduchom pseudokóde?
Princípu rozumiem, avšak niektoré implementacné detaily mi nie su zjavné.
Napríklad, pri Queue-based Load Levelingu je jedna z výhod to, že namiesto timeoutovania klienta sa z





Implemetations

https://github.com/AndyButland/QueueBasedLoadLevellingDemo

https://github.com/riserrad/queue-based-load-leveling/blob/master/src/console/Program.cs