Resilience4j

Features

- Lightweight fault tolerance library inspired by Hystrix, but designed for functional programming
- Uses Vavr, functional library for Java 8+ that provides persistent data types & functional control structures
- Provides higher-order functions (decorators) to enhance any funtional interface, lambda expression or method reference with a :

    A. Circuit Breaker

    B. Bulkhead

    C. Retry

    D. Rate Limiter

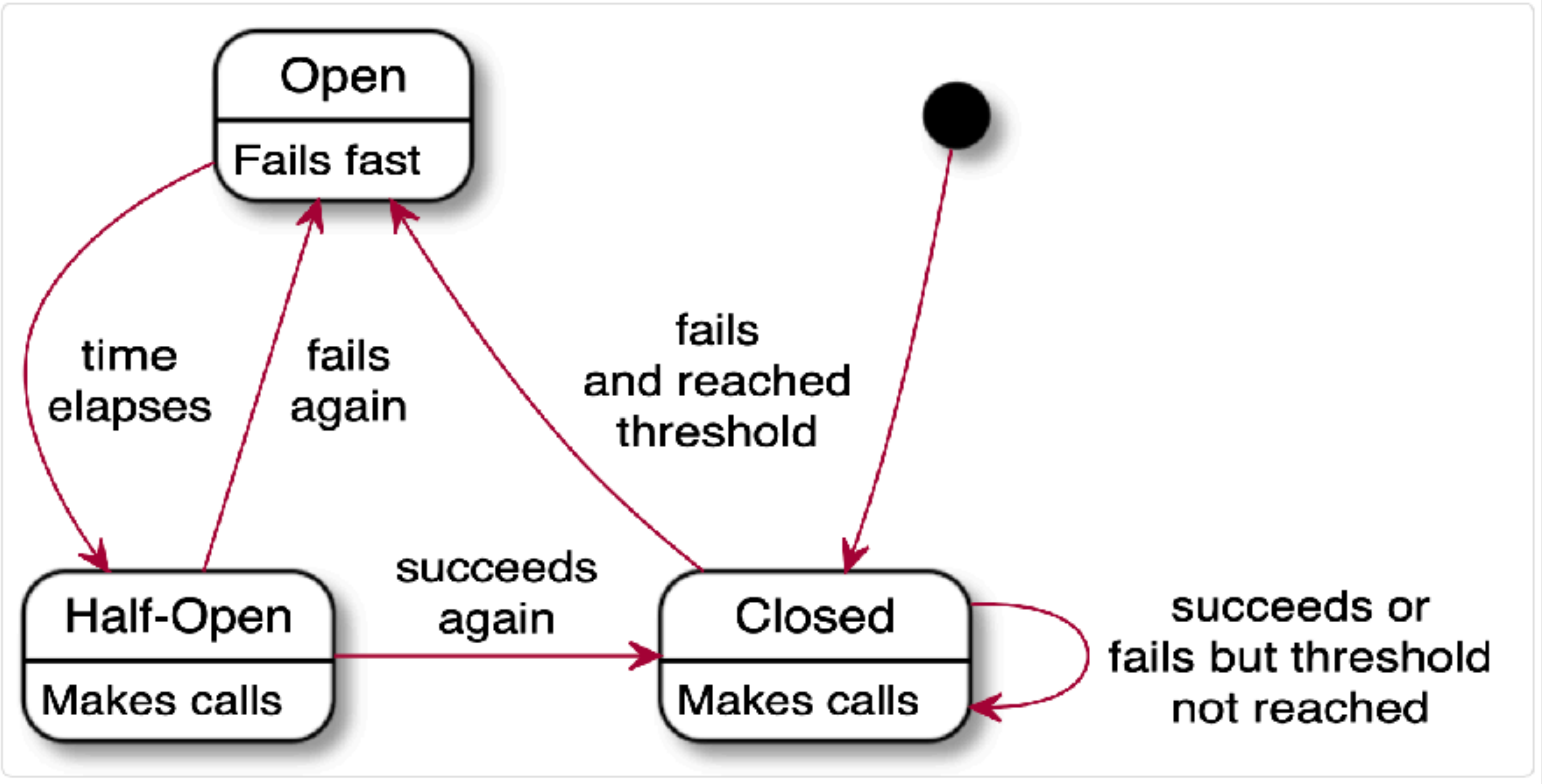
    E. Time Limiter

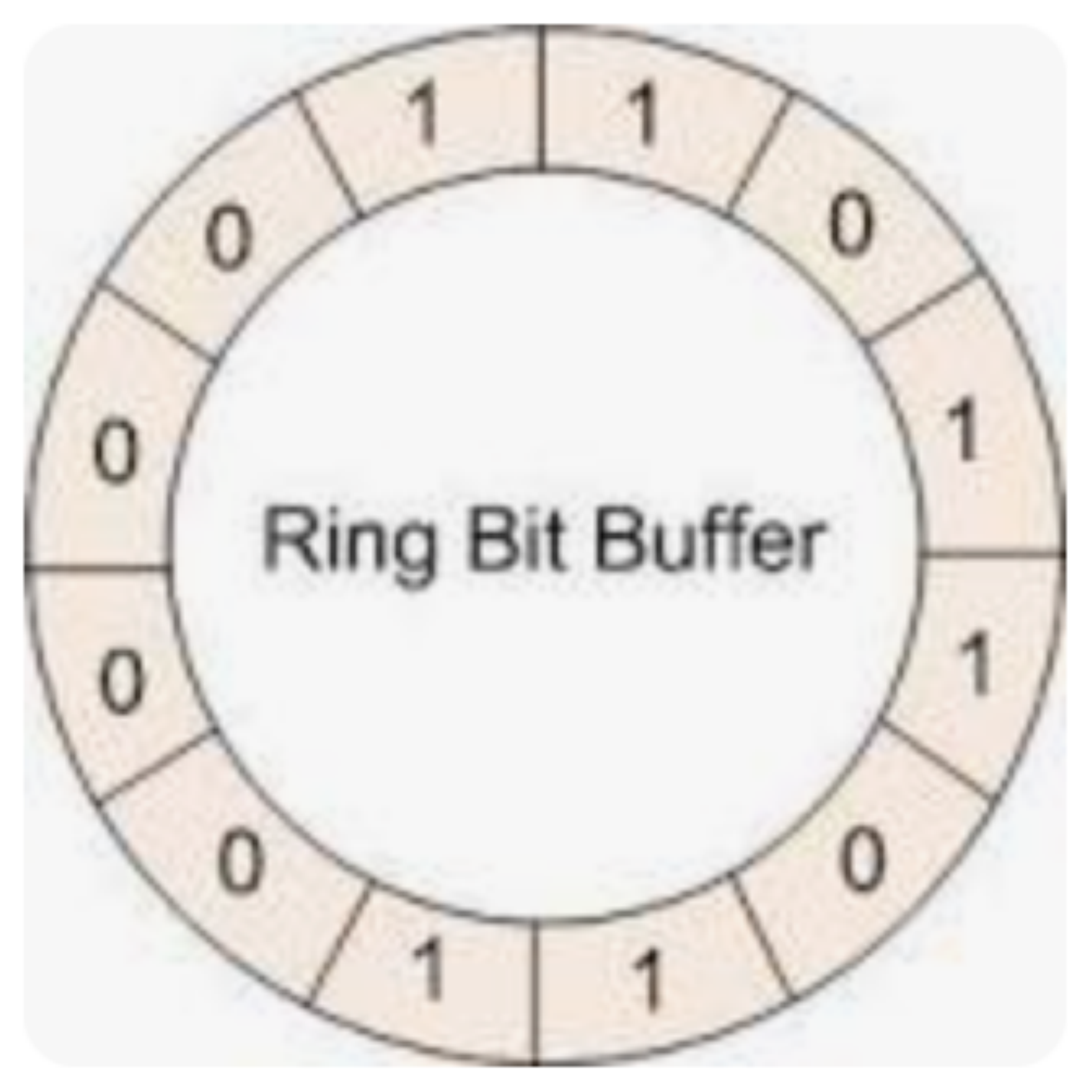
    F. Cache

Circuit Breaker

- Uses window based approach to calculate failure rate based on time or count
- Opens when too many slow calls are detected above response time threshold in the window, saving the slower service from being bombarded with requests giving it time to recover
- Enables to configure number of calls to be performed in HALF_OPEN state increasing accuracy while closing the circuit
- Kinds of exceptions to be considered relevant & irrelevant is also configurable
- Can be configured to go from OPEN to HALF_OPEN automatically

- Uses Ring Bit Buffer in the CLOSED state to store success or failure statuses of functional calls





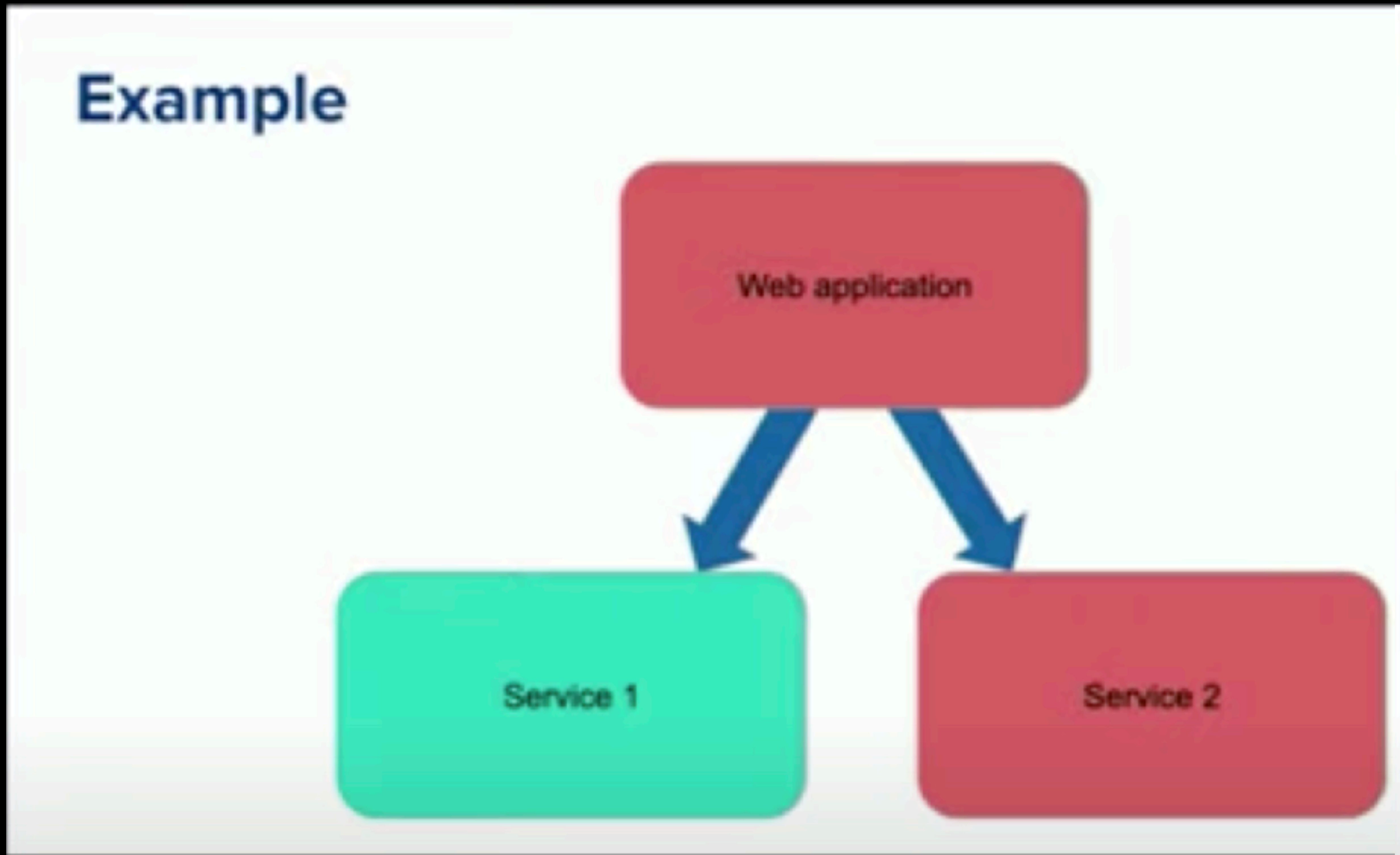
- 5 states : CLOSED , OPEN , HALF_OPEN , DISABLED (Always allow) , FORCED_OPEN (Always Deny)

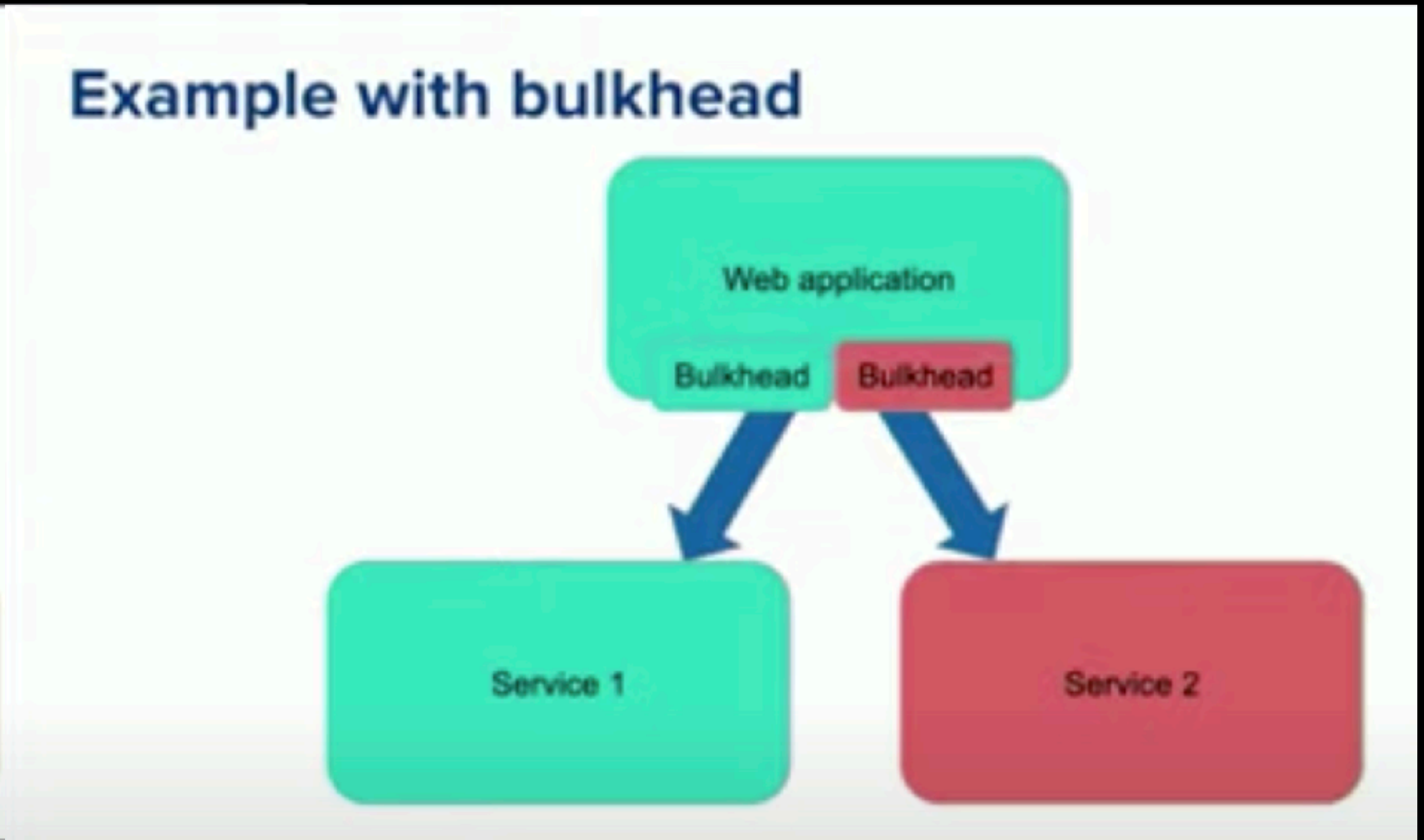
Bulkhead

- Resilience4j provides 2 implementations to limit number of concurrent execution :

  A. SemaphoreBulkhead uses semaphores

  B. FixedThreadPoolBulkhead uses bounded queue and fixed thread pool





Retry

- Number of times call made to 3rd party to check if its health is good or not, if good then close the circuit else leave it open and keep retrying
- Can be configured as number of requests per second
- Backoff strategy & multiplier can be configured
- Use cases could be :
    A. For dependencies that are sporadically unstable
    B. Won't use for GET, but for POST/PUT
    C. Need to handle calls exceeding maxAttempts

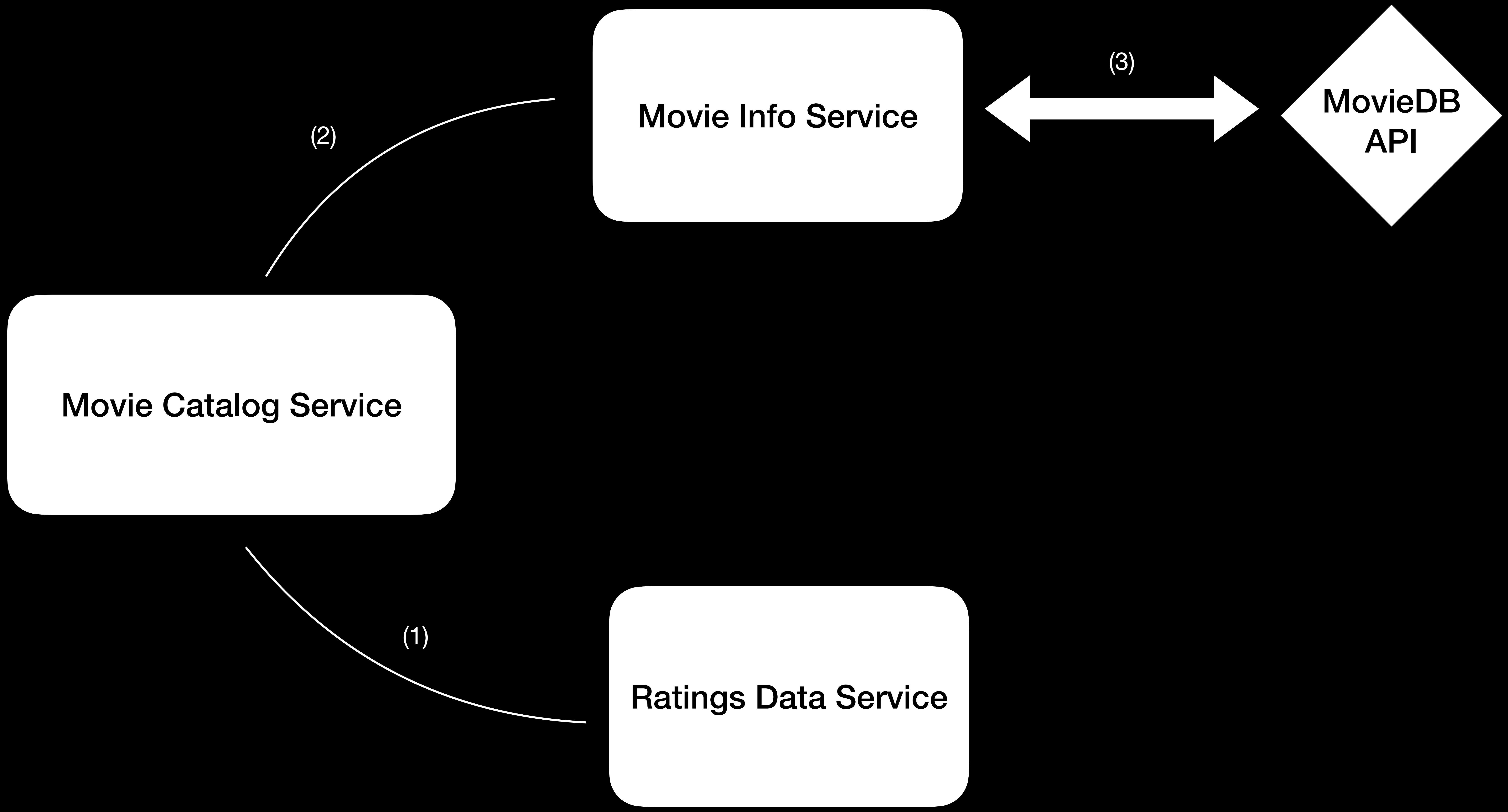
RateLimiter

- To control rate of traffic sent or received by network interface controller
- To prevent DoS attacks
- Can configure requests per second like n such that if there are more than n requests per sec then only n will be honoured and rest will be discarded

References

- https://resilience4j.readme.io/docs/comparison-to-netflix-hystrix
- https://bit.ly/3oZY0bv
- https://www.javaadvent.com/2020/12/resilient-applications-spring-resilience4j.html
- https://www.baeldung.com/resilience4j
- https://www.baeldung.com/spring-cloud-circuit-breaker
- https://youtu.be/NHVxrLb3jFI

Movie Catalog Architecture


(3)
MovieDB
API
Movie Info Service
(2)
Movie Catalog Service
(1)
Ratings Data Service