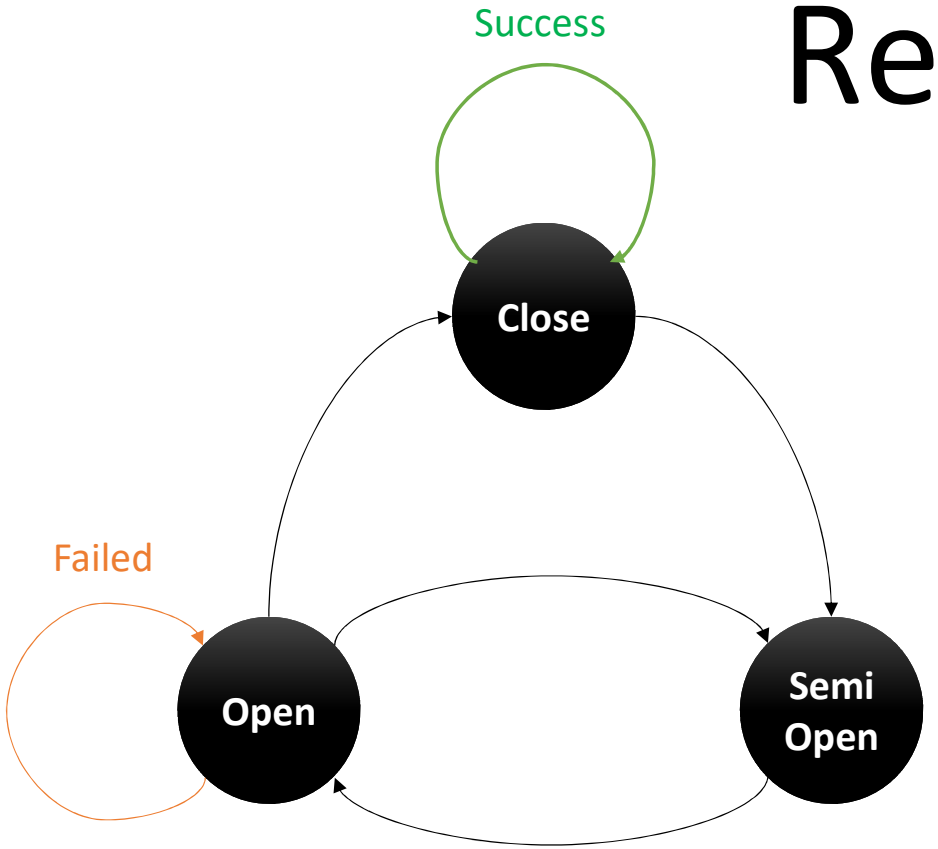
Circuit Breaker et Resilience4j

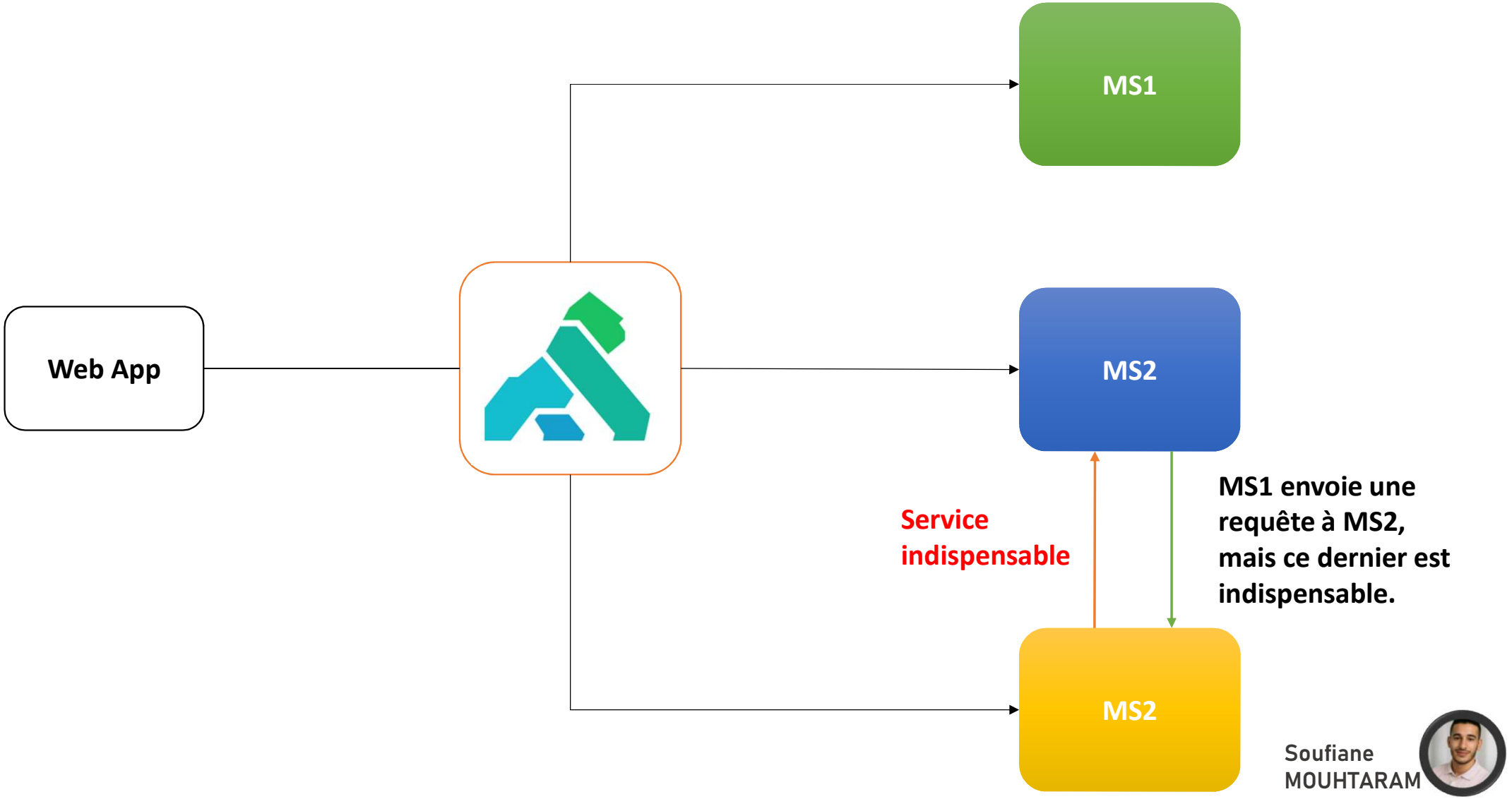


RESILIENCE4J

spring

Soufiane MOUHTARAM

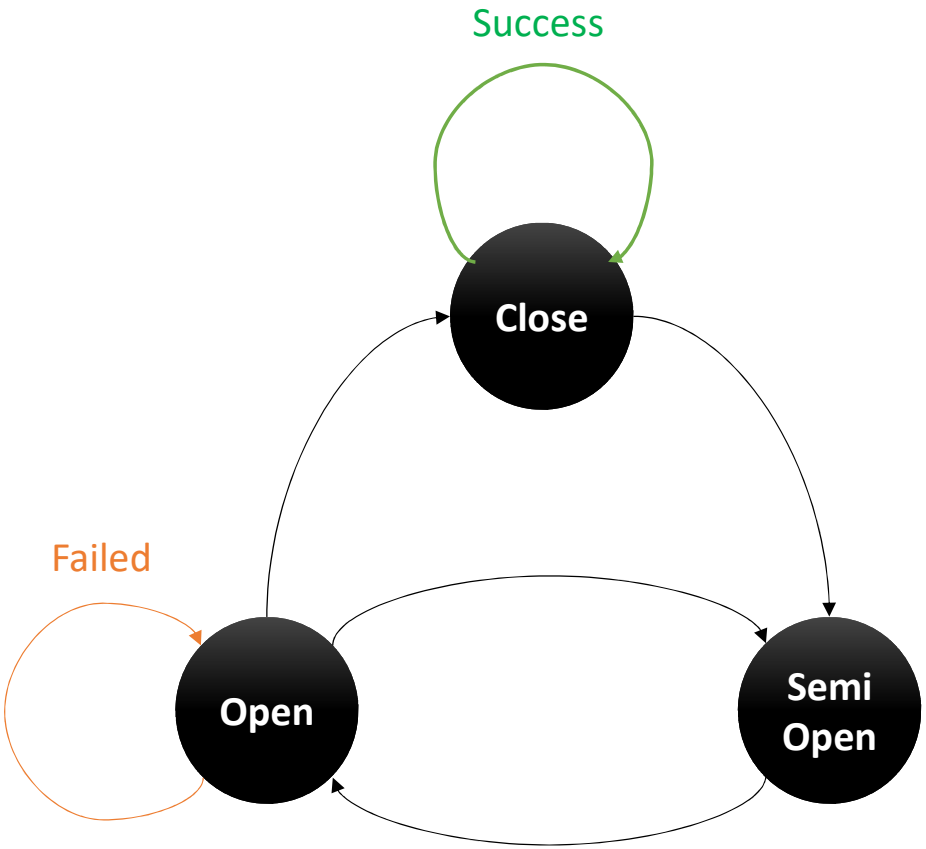
Circuit Breaker



Circuit Breaker

Lorsqu'un service est **indispensable** et qu'il n'est pas disponible au moment où une requête lui est adressée, il est crucial de garantir que cette requête soit bien prise en compte dès que le service redevient accessible.

C'est précisément le rôle du pattern **Circuit Breaker**.
Ce dernier permet de gérer les défaillances temporaires d'un service en évitant d'effectuer des appels inutiles lorsque le service est indisponible.
Le fonctionnement est le suivant :
1- **État fermé (Closed)** : Les requêtes sont transmises normalement au service.
2- Si plusieurs échecs sont détectés (par exemple 3 tentatives échouées), le circuit passe à l'état semi-ouvert (Half-Open).
3- En état **semi-ouvert**, le système effectue un nombre limité de tentatives pour tester si le service est de nouveau fonctionnel.
4- Si les tentatives échouent encore, le circuit passe en état **ouvert (Open)** : aucune requête n'est transmise pendant un certain délai d'attente.
5- Après ce délai, une nouvelle tentative est effectuée. Si elle réussit, le circuit repasse à l'état fermé, et les appels reprennent normalement.



Soufiane
MOUHTARAM


Les avantages de Circuit Breaker

Protège le système contre les pannes en cascade

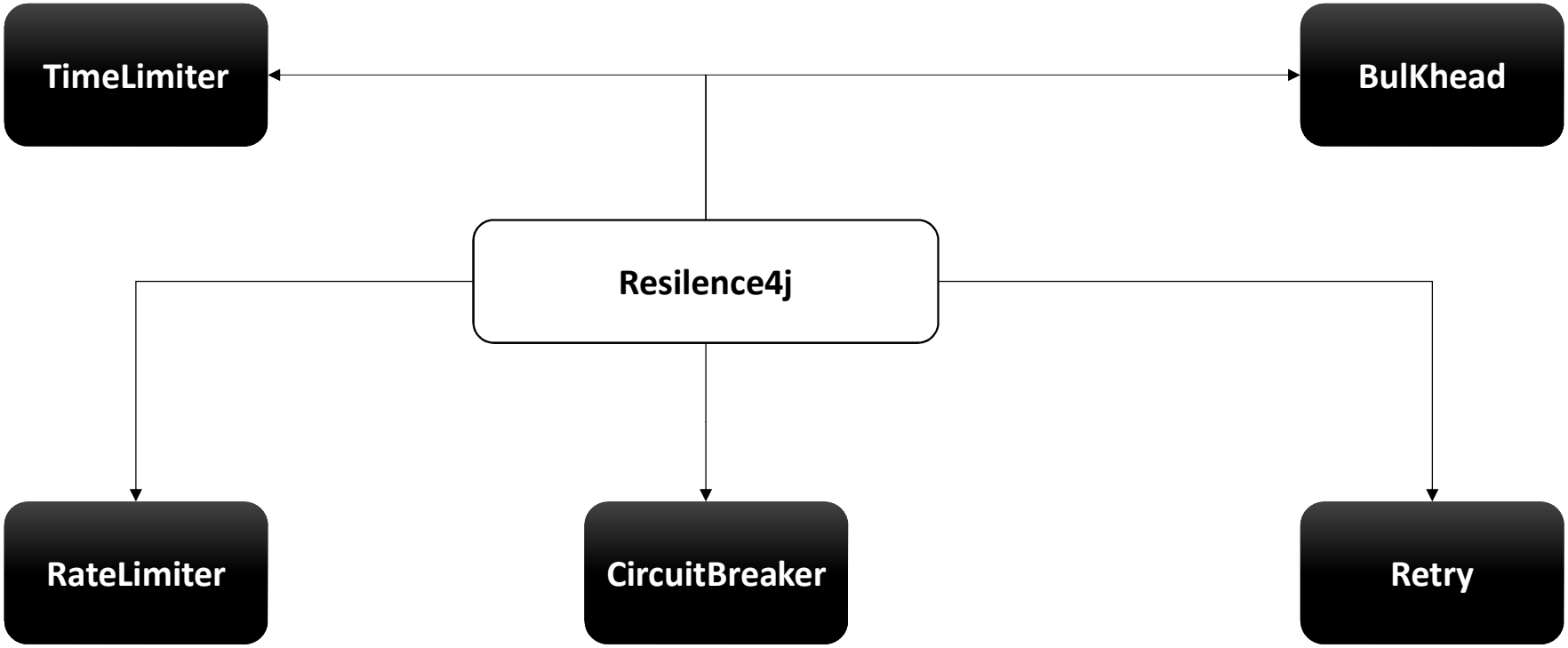
Facilite le monitoring

Facile à configurer

Reprise automatique

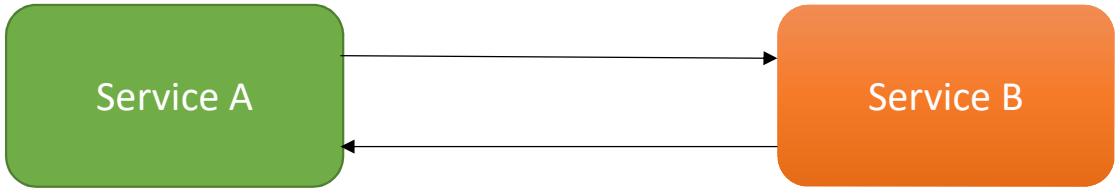
Soufiane
MOUHTARAM

Resilience4j est une bibliothèque Java qui fournit une implémentation complète du **pattern Circuit Breaker**, ainsi que d'autres patterns de résilience utiles dans les architectures **microservices**.



Soufiane
MOUHTARAM

Lorsque l'appel du service A vers le service B échoue, il effectue **3 tentatives** avec une pause de **2 secondes** entre chaque essai.




```
resilience4j:
  retry:
    instances:
      serviceB:
        max-attempts: 3
        wait-duration: 2s
  circuitbreaker:
    instances:
      serviceB:
        register-health-indicator: true
        sliding-window-size: 5
        failure-rate-threshold: 50
        wait-duration-in-open-state: 10s
```

Instance de retry pour le service B

Nombre maximal de tentatives en cas d'échec (3 essais)

Durée d'attente entre chaque tentative (2 secondes)

Instance de circuit breaker pour serviceB

Active le suivi de santé dans Spring Actuator

Taille de la fenêtre glissante pour calculer le taux d'échec (5 appels)

Seuil de taux d'échec (%) pour ouvrir le circuit (50%)

Durée pendant laquelle le circuit reste ouvert avant de retenter (10 secondes)

Soufiane MOUHTARAM