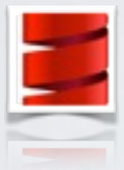
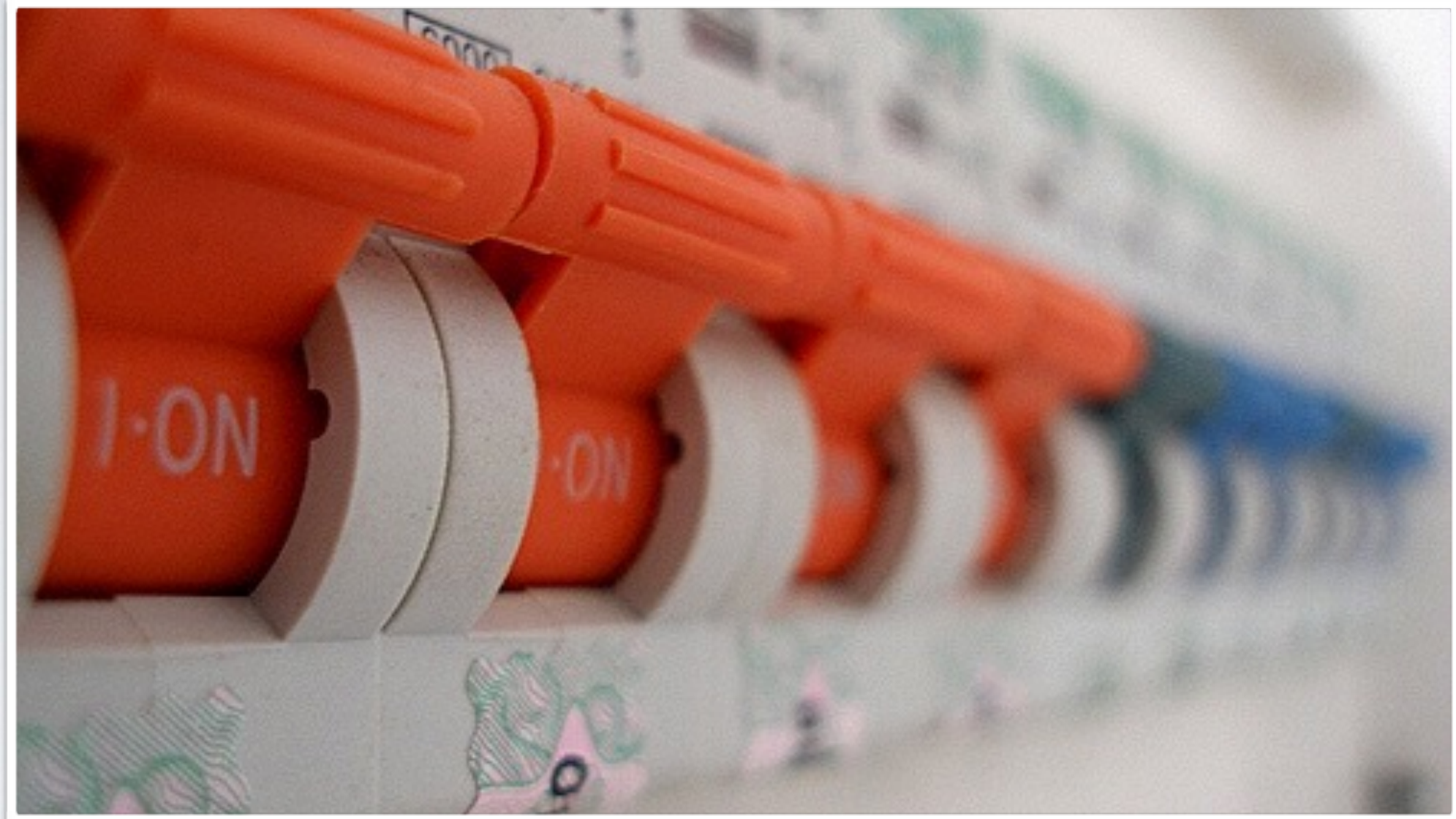
Tolérance Aux Pannes Avec LE CIRCUIT BREAKER PATTERN

Mouhcine Moulou

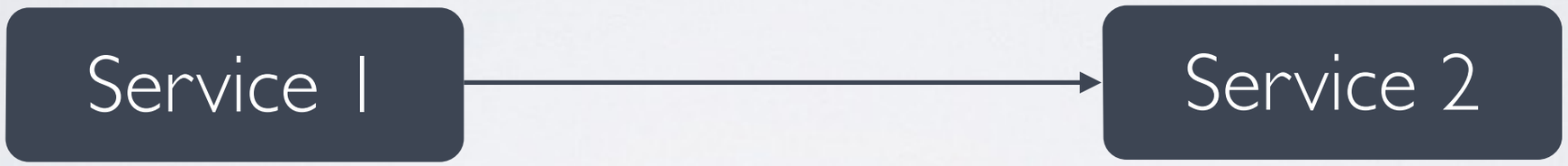

@mouloumouhcine


Consultant Scala

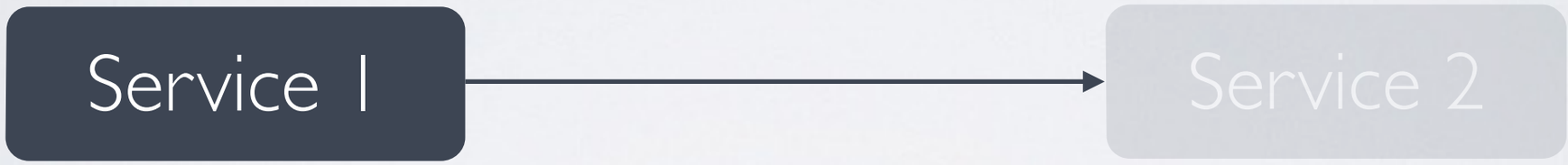
SOAT


ON
ON

PROBLÈME



PROBLÈME



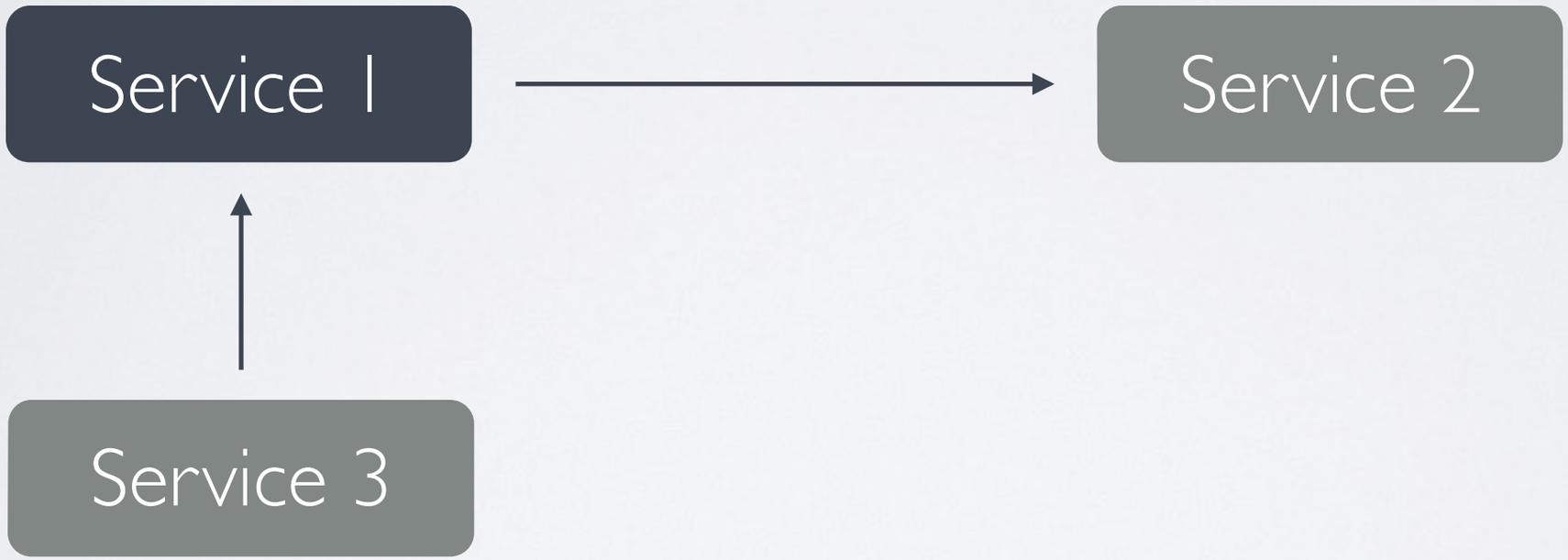
Gaspillage des Resources VM (Thread, Mémoire, etc.)

PROBLÈME

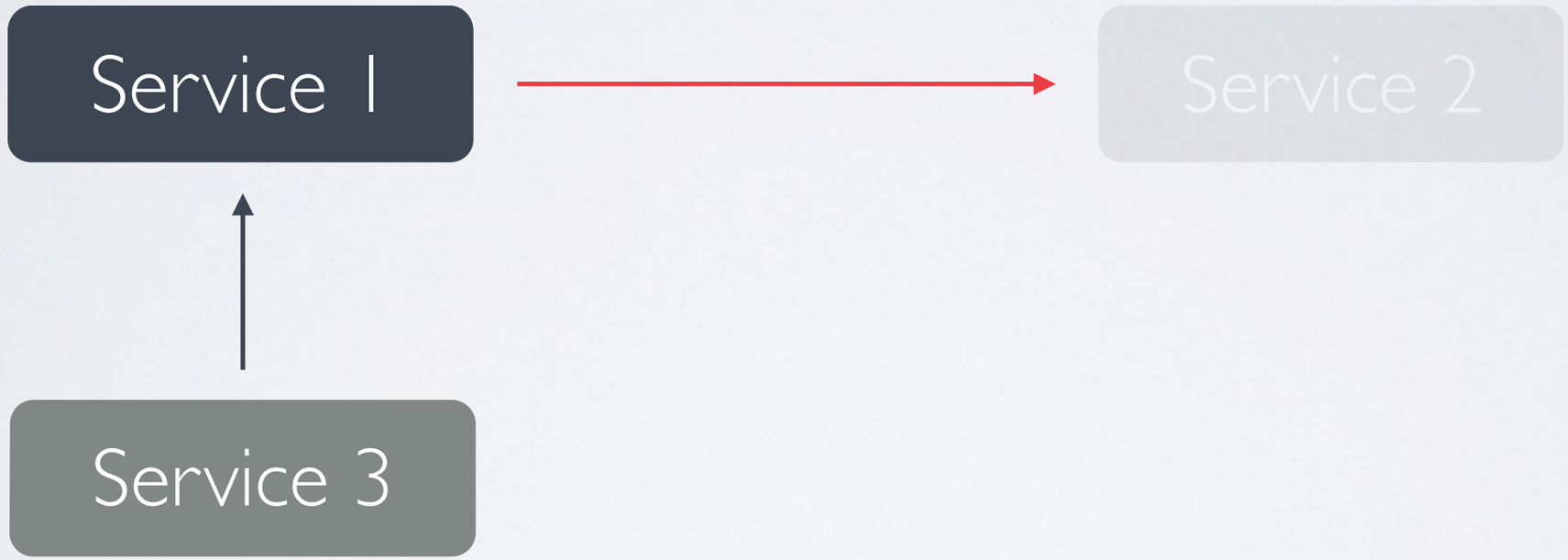
PROBLÈME

Performance ↘

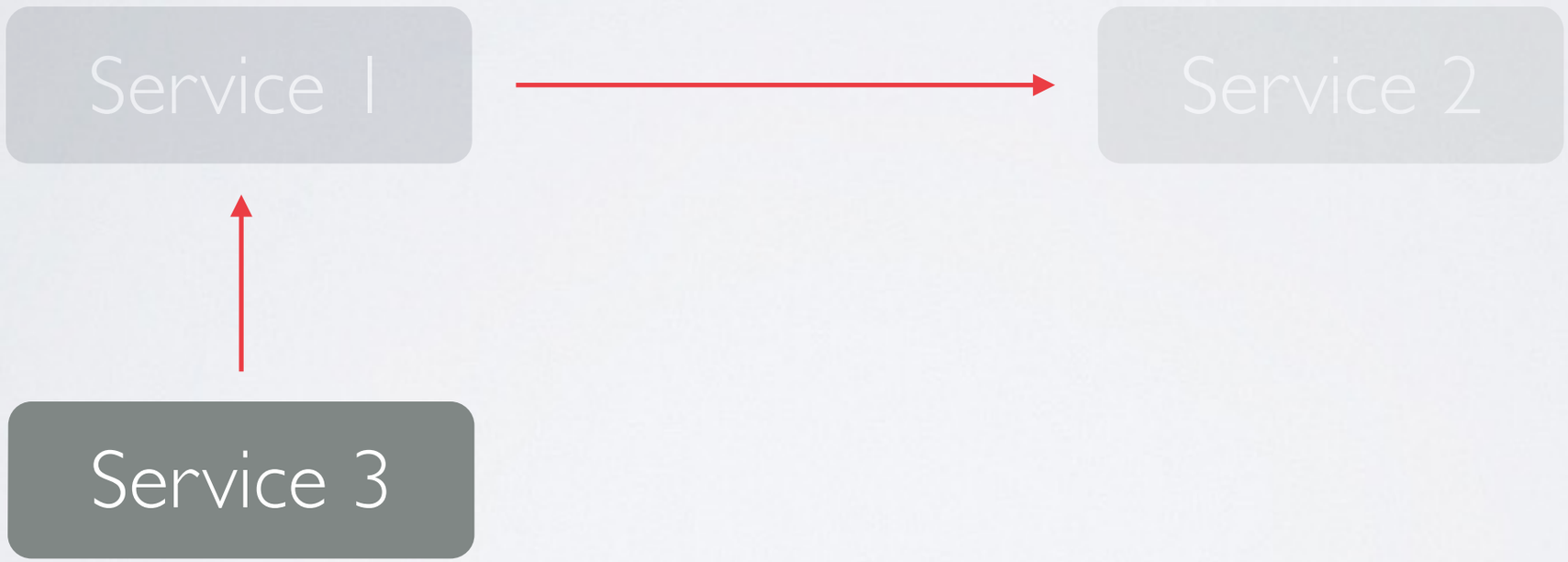
PROBLÈME



PROBLÈME



PROBLÈME



PROBLÈME

CASCADING FAILURES

SOLUTION

SOLUTION


A problem has been detected and windows has been shut down to prevent damage
to your computer.
The problem seems to be caused by the following file: SPCMDCON.SYS
PAGE_FAULT_IN_NONPAGED_AREA
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:
_LET_IT_CRASH_
If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.
Technical information:
*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)
*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c

« Let It Crash & Handle with Grace. »

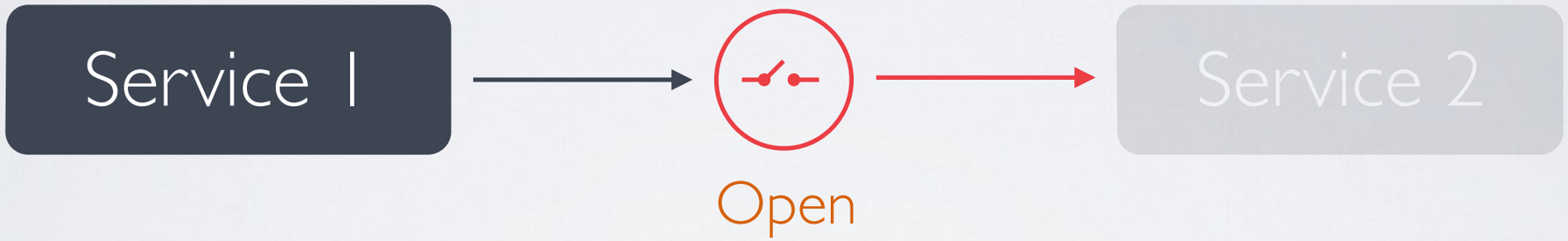
CIRCUIT BREAKER

CIRCUIT BREAKER

CIRCUIT BREAKER

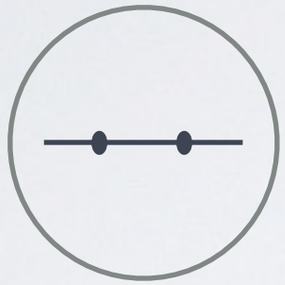
Closed

CIRCUIT BREAKER



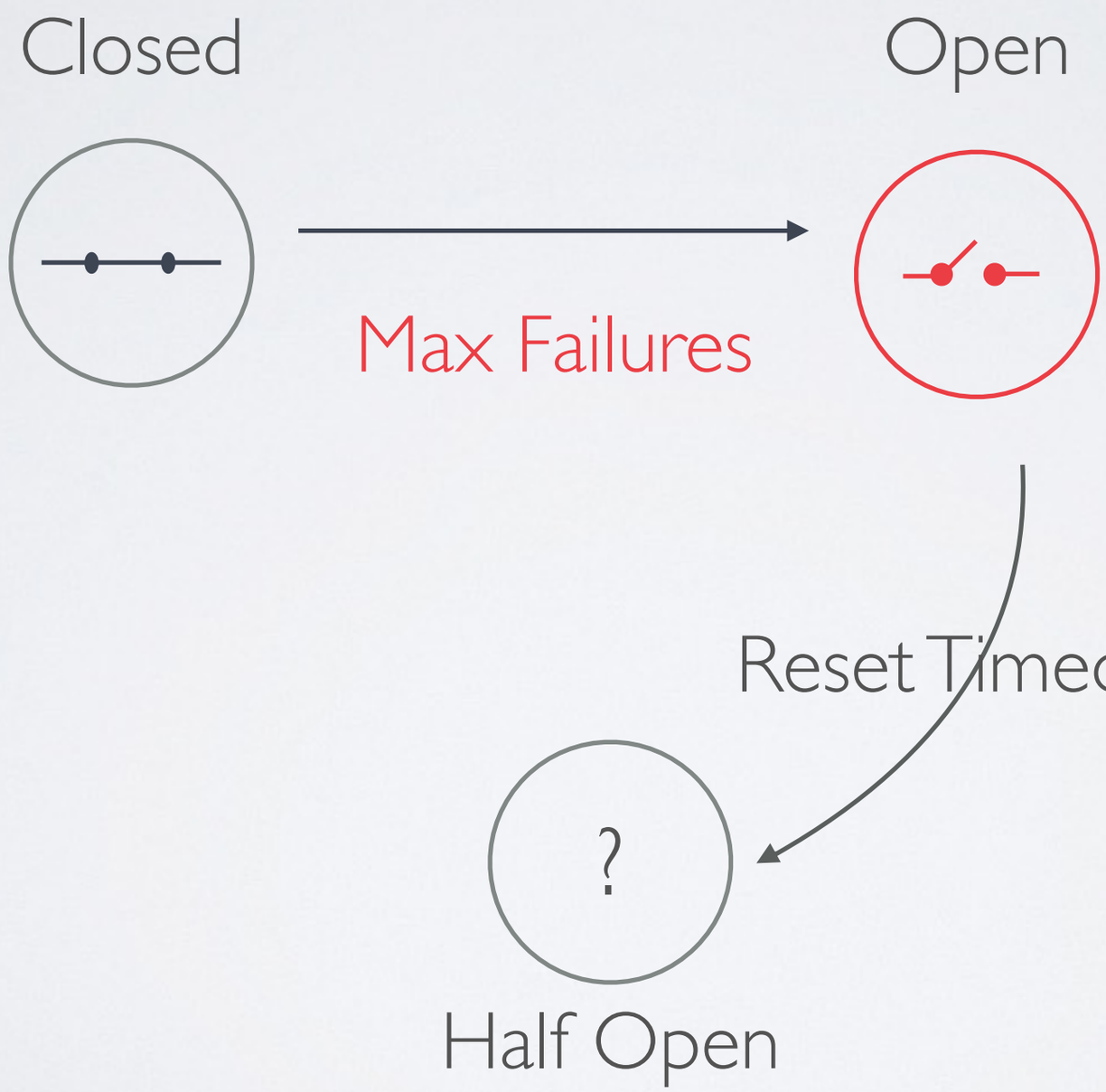
CIRCUIT BREAKER

Closed

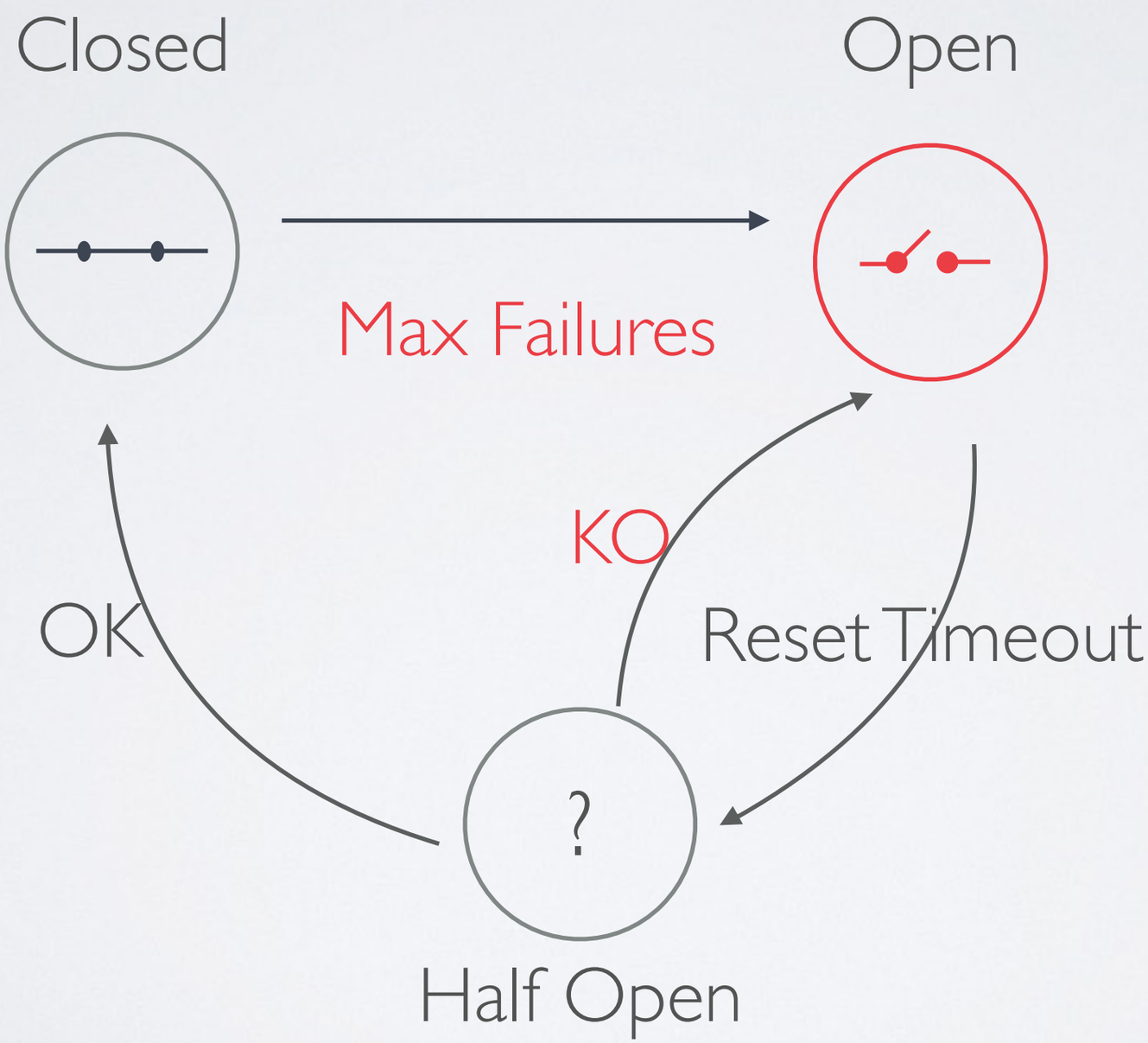


CIRCUIT BREAKER

CIRCUIT BREAKER



CIRCUIT BREAKER



EXEMPLE (AKKA IMPLEMENTATION)

```
CircuitBreaker breaker = new CircuitBreaker(
        getContext().dispatcher(), // L'actor System
        getContext().system().scheduler(), // Scheduler
        50, // maxFailures
        Duration.create(10, "s"), // Call Timeout
        Duration.create(1, "m") // Reset Timeout
);
```

EXEMPLE (AKKA IMPLEMENTATION)

```
// Sans Circuit Breaker

return dangerousCall();
```

EXEMPLE (AKKA IMPLEMENTATION)

```
// Callable
new Callable<String>() {
    public String call() throws Exception {
        return dangerousCall();
    }
}
```

EXEMPLE (AKKA IMPLEMENTATION)

```
// Avec Circuit Breaker
// Appel Java
breaker.callWithSyncCircuitBreaker(
        new Callable<String>() {
            public String call() throws Exception {
                return dangerousCall();
            }
        }
);
```

EXEMPLE (AKKA IMPLEMENTATION)

```
// Avec Circuit Breaker
// Appel Scala
breaker.callWithCircuitBreaker(
    Future( dangerousCall())
)
```

CIRCUIT BREAKER

Avantages

- Réponse rapide.
- Meilleur utilisation des resources.
- Cascading failures.
- Monitoring.

CIRCUIT BREAKER

Avantages

- Réponse rapide.
- Meilleur utilisation des resources.
- Cascading failures.
- Monitoring.

Inconvénients

- Impossibilité de filtrer les exceptions

CIRCUIT BREAKER

Avantages

- Réponse rapide.
- Meilleur utilisation des resources.
- Cascading failures.
- Monitoring.

Inconvénients

- Impossibilité de filtrer les exceptions

ESSAYEZ LE !