

## MICROSERVICES DESIGN PATTERNS

### CIRCUIT BREAKER PATTERN

→ It addresses the challenge of preventing cascading failures.

For example, network or service failure in one microservice can quickly cascade into other services and cause a system-wide failure.

→ Why We Need Circuit Breaker Pattern?

Services have to communicate with each other. Sometimes, there can be service failures, or it takes significant time to respond to requests due to network connectivity issues.

A microservice architecture consists of a large number of interconnected microservices. So, a failure of a single microservice can have a cascading effect on all microservices and can significantly affect system availability.

Hence, developers needed a method to prevent this cascading effect between microservices in case of network connectivity issues, timeouts, or service unavailability. The circuit breaker pattern was introduced to address these issues.

→ What is the Circuit Breaker Pattern?

The circuit breaker pattern falls under the sustainable design patterns category. It allows developers to prevent cascading failures in microservices architecture by invoking remote services through a proxy.

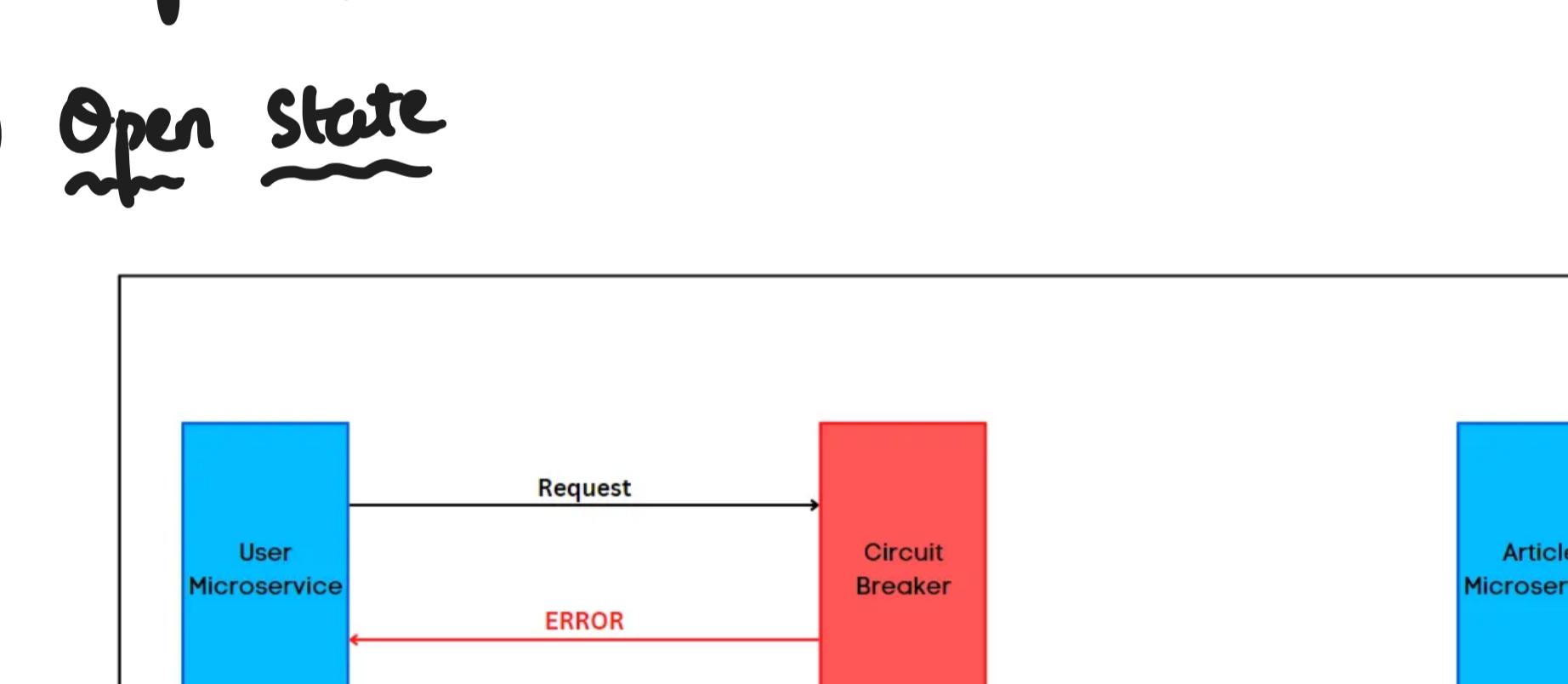
In simple terms, the circuit breaker pattern is a fail-safe mechanism to prevent cascading failures of microservices.

→ How does Circuit Breaker Pattern Work?

It has three states:

- ① Closed
- ② Open
- ③ Half Open

#### 1) Closed State

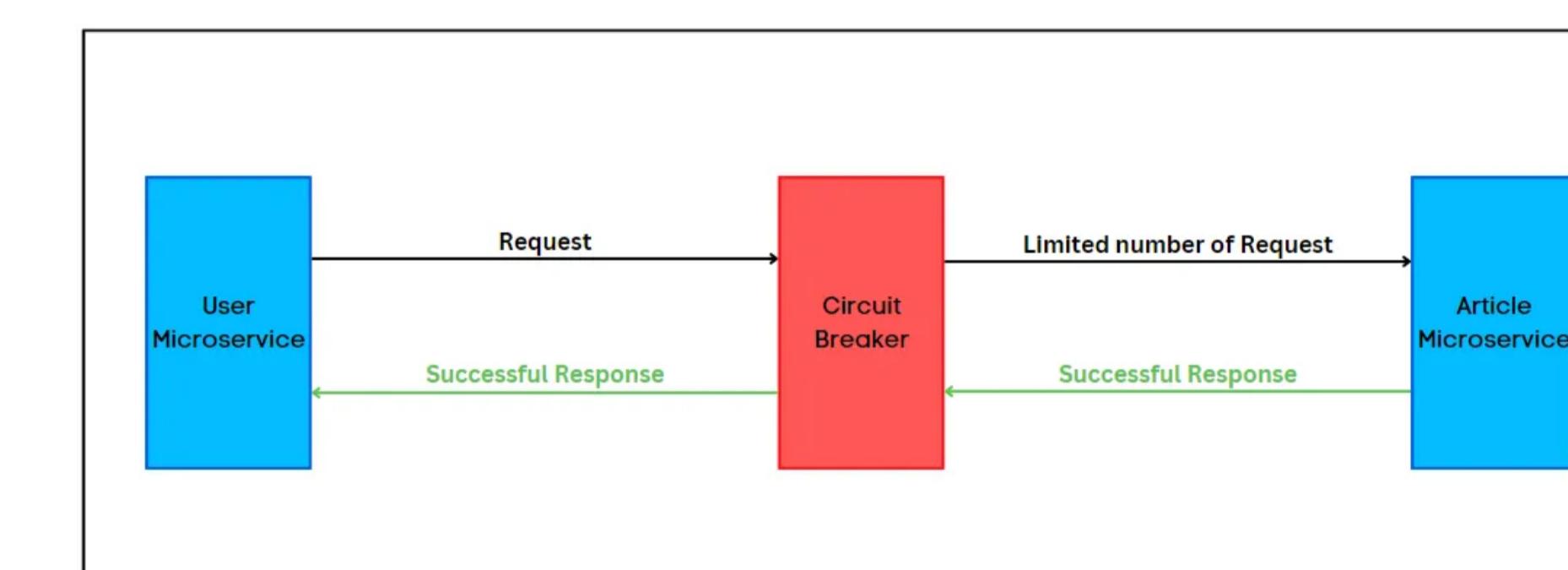


The initial state of the circuit breaker proxy is the Closed state. The CB allows microservices to communicate as usual and monitor the number of failures occurring within the defined time period.

If the failure count exceeds the specified threshold value, the circuit breaker will move to Open state.

If not, it will reset the failure count & timeout period.

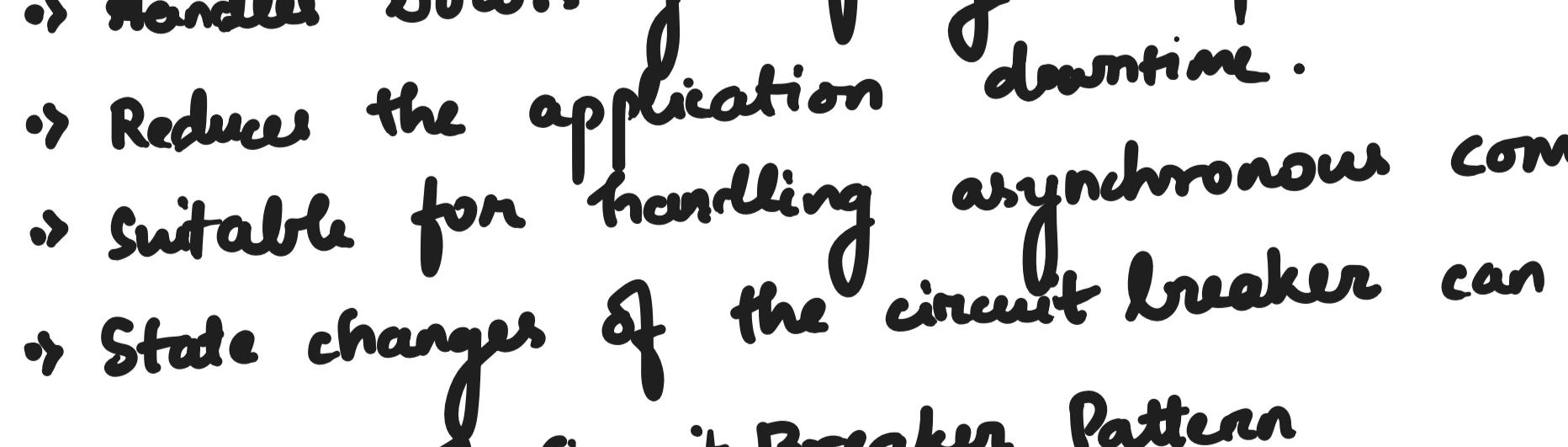
#### 2) Open State



Once the CB moves to the Open state, it will completely block the communication between microservices. So, the article service will not receive any requests, and the user service will receive an error from the circuit breaker.

The CB will remain in the Open state until the timeout period ends. Then it will move into the Half-Open state.

#### 3) Half-Open State



In the Half-Open state, the circuit breaker will allow a limited number of requests to reach article service.

If those requests are successful, the circuit breaker will switch the state to Closed and allow normal operations. If not, it will again block the requests for the defined timeout period.

→ Advantages of Circuit Breaker Pattern

- Helps to prevent cascading failure
- Handles errors gracefully and provides better user experience
- Reduces the application downtime.
- Suitable for handling asynchronous communication
- State changes of the circuit breaker can be used for error monitoring.

→ Challenges of Circuit Breaker Pattern

- Throughput issues in services if not properly configured
- Need good infrastructure management to maintain circuit breakers
- Difficult to test