

Andreas Prüller

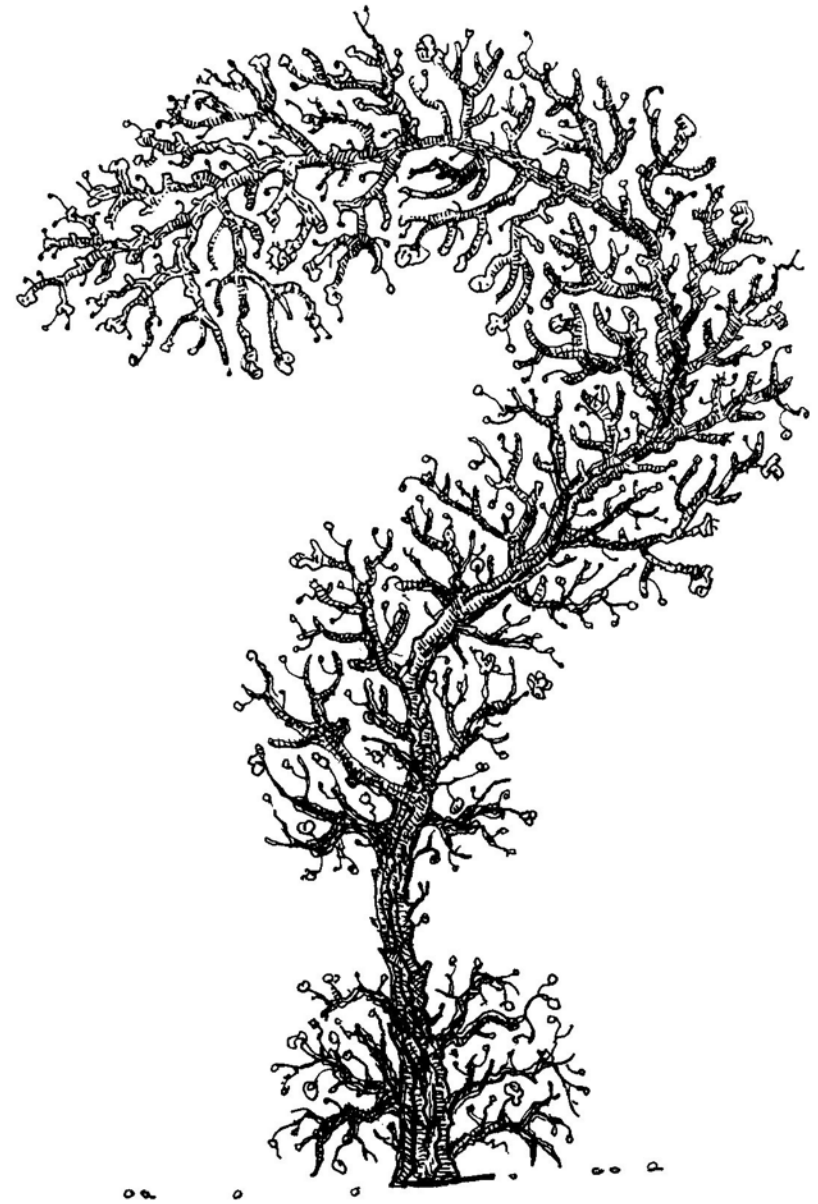
Unknown Patterns

Circuit Breaker
Pattern

Unknown Patterns

The Circuit Breaker Pattern

The Problem

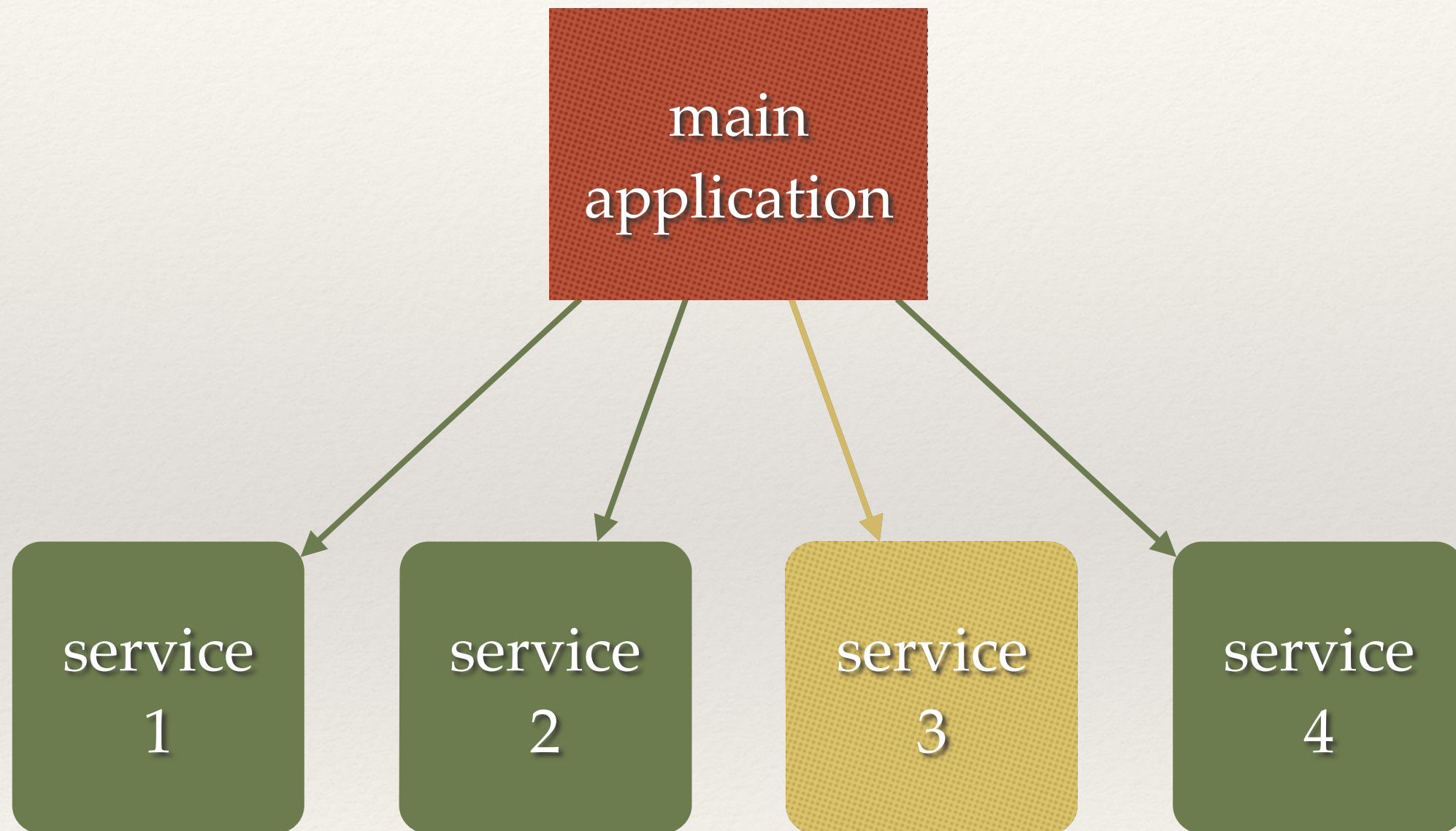


Don Moyer: <http://www.flickr.com/photos/36106576@N05/4230355503/>

The Problem

- ❖ Remote service calls to dying (or even worse: dead) system take down the main application.
- ❖ The more remote calls used, the higher the risk to die with one of them.

The Problem



Unknown Patterns

The Circuit Breaker Pattern

The Solution



The Circuit Breaker Pattern

What to do?

- ❖ detect failure on remote call
- ❖ in case of failure prevent further calls
- ❖ need for retry mechanism to notice when remote system is up again

The Circuit Breaker Pattern

3 states: OPEN, CLOSED, HALF_OPEN

❖ CLOSED

Connections to the monitored subsystem are passed through as normal

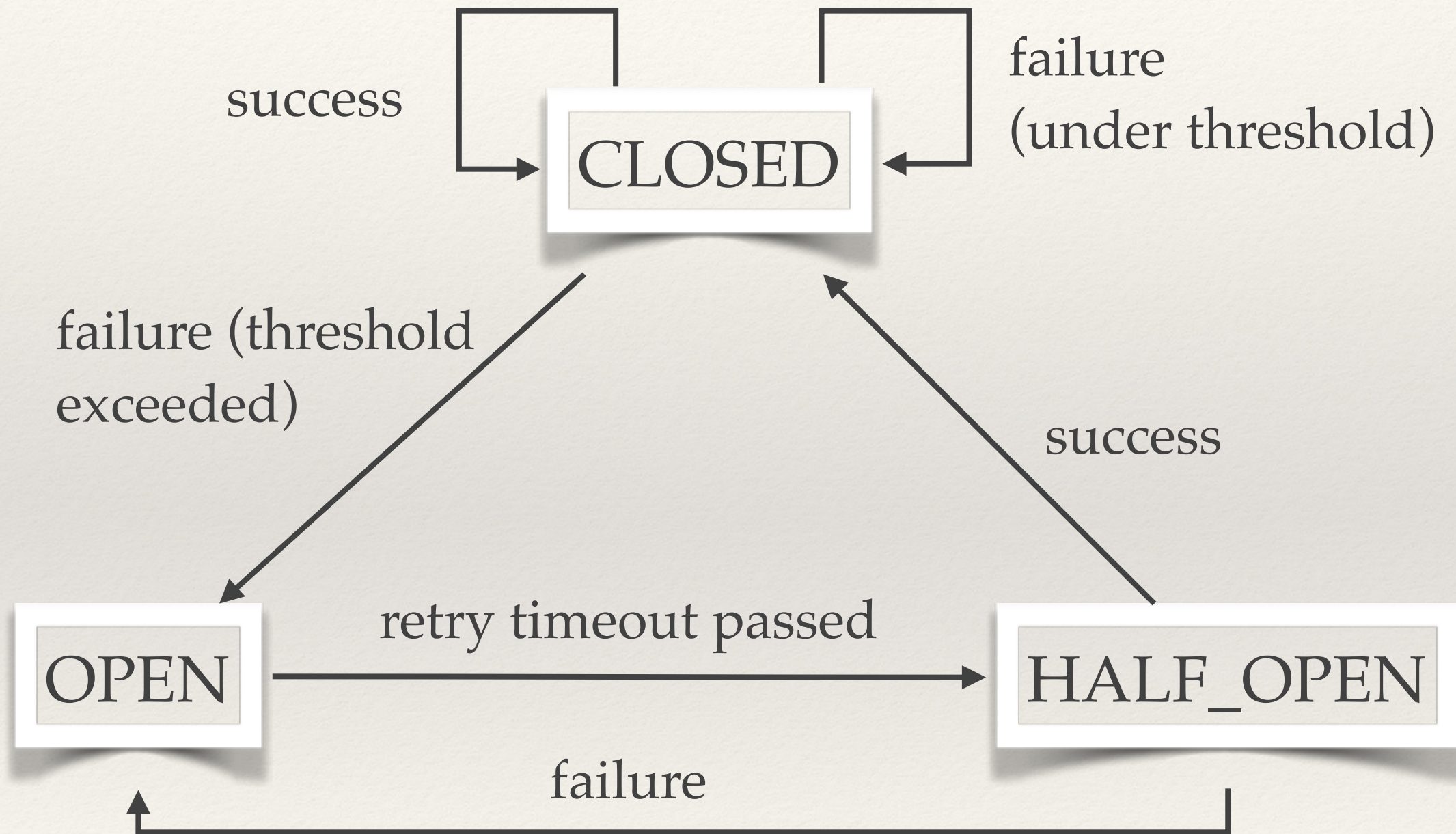
❖ OPEN

Connections to the monitored subsystem are intercepted and immediately fail

❖ HALF_OPEN

A limited number of connections to the monitored subsystem are allowed to pass through but further failures lead is right back to OPEN

The Circuit Breaker Pattern



The Circuit Breaker Pattern

Considerations on Implementation

- ❖ Multi threaded environment
- ❖ Performance
- ❖ Manual override
- ❖ Monitoring

The Circuit Breaker Pattern

Further Reading

- ❖ <http://martinfowler.com/bliki/CircuitBreaker.html>
- ❖ <http://dev.hubspot.com/blog/bid/64543/Building-a-Robust-System-Using-the-Circuit-Breaker-Pattern>
- ❖ <http://techblog.netflix.com/2011/12/making-netflix-api-more-resilient.html>
- ❖ <http://code.google.com/p/jianwikis/wiki/CircuitBreakerDesignPattern>
- ❖ Michael T. Nygard: Release It!: Design and Deploy Production-Ready Software, 2007, ISBN: 978-0978739218

The Circuit Breaker Pattern

Implementations

- ❖ <https://github.com/Netflix/Hystrix>
- ❖ <https://github.com/Comcast/jrugged>
- ❖ Spring Integration:
RequestHandlerCircuitBreakerAdvice

Thank you!