

# Kresil

# Kotlin Resilience

Kotlin Multiplatform library for fault-tolerance  
with Ktor Integration

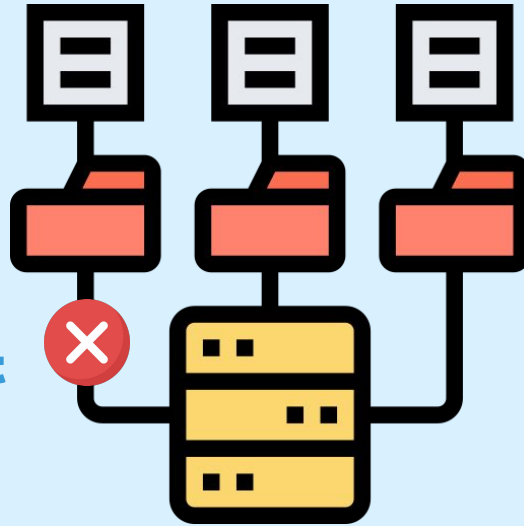


**ISEL**  
INSTITUTO SUPERIOR DE  
ENGENHARIA DE LISBOA

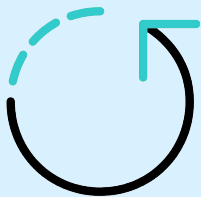
**Supervisor: Prof. Pedro Félix**  
**Author: Francisco Engenheiro - 49428**  
**Project and Seminary**  
**BSc in Computer Science and Engineering**  
**Summer 2023/2024**

# Resilience in Distributed Systems

If failure is inevitable, what can be done?



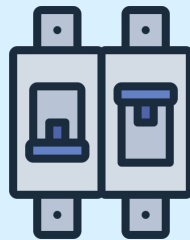
# Resilience Mechanisms



**Retry** - Repeats failed executions



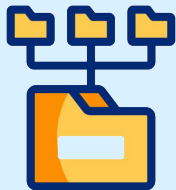
**Rate Limiter** - Limits executions/period



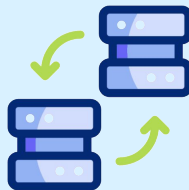
**Circuit Breaker** - Temporarily blocks possible failures



**Time Limiter** - Limits duration of execution



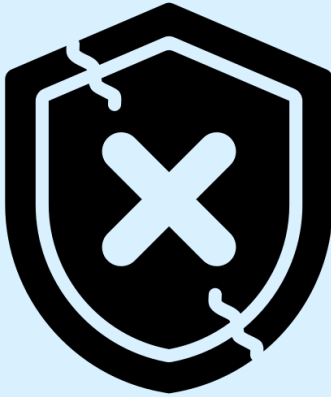
**Cache** - Memorizes a successful result



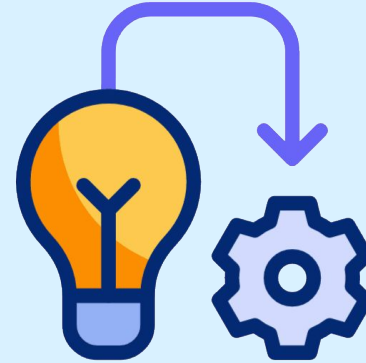
**Fallback** - Defines an action to fallback on failure

And many more...

# Resilience Types

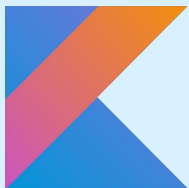


**Reactive** - Mitigates  
impact from failures



**Proactive** - Prevents  
failures from happening

# Technologies



## Kotlin Multiplatform

Allows sharing code  
across multiple platforms

Kotlin/JS



Kotlin/Native



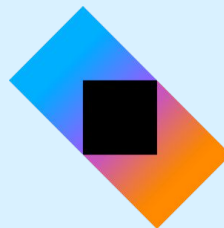
Kotlin/Android



Kotlin/JVM



# Frameworks



## Ktor

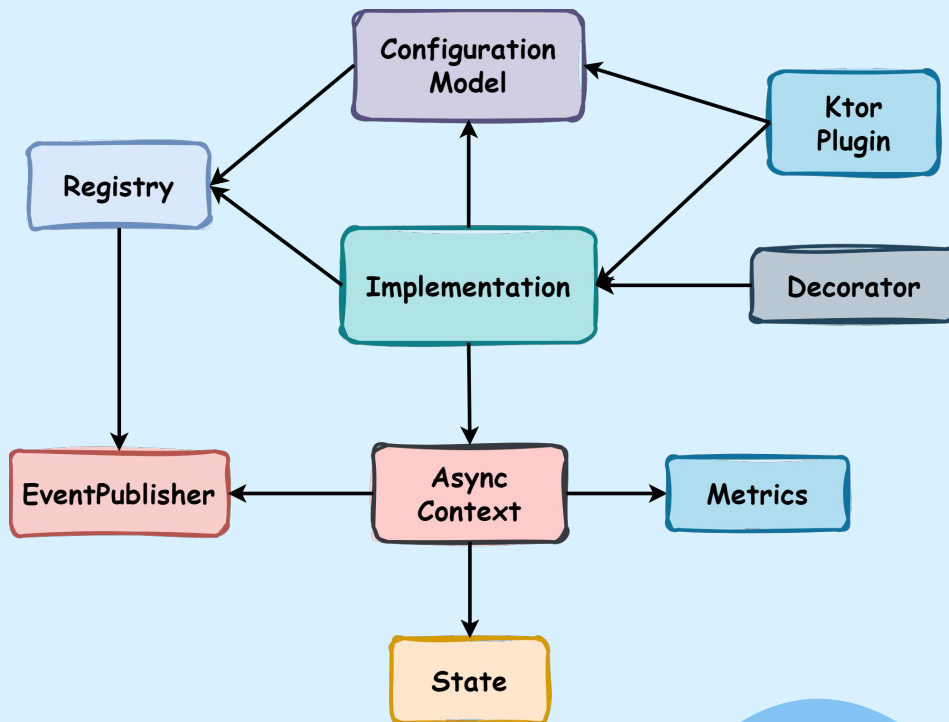
Asynchronous Kotlin  
Multiplatform server  
and client development

# Mechanism Configuration

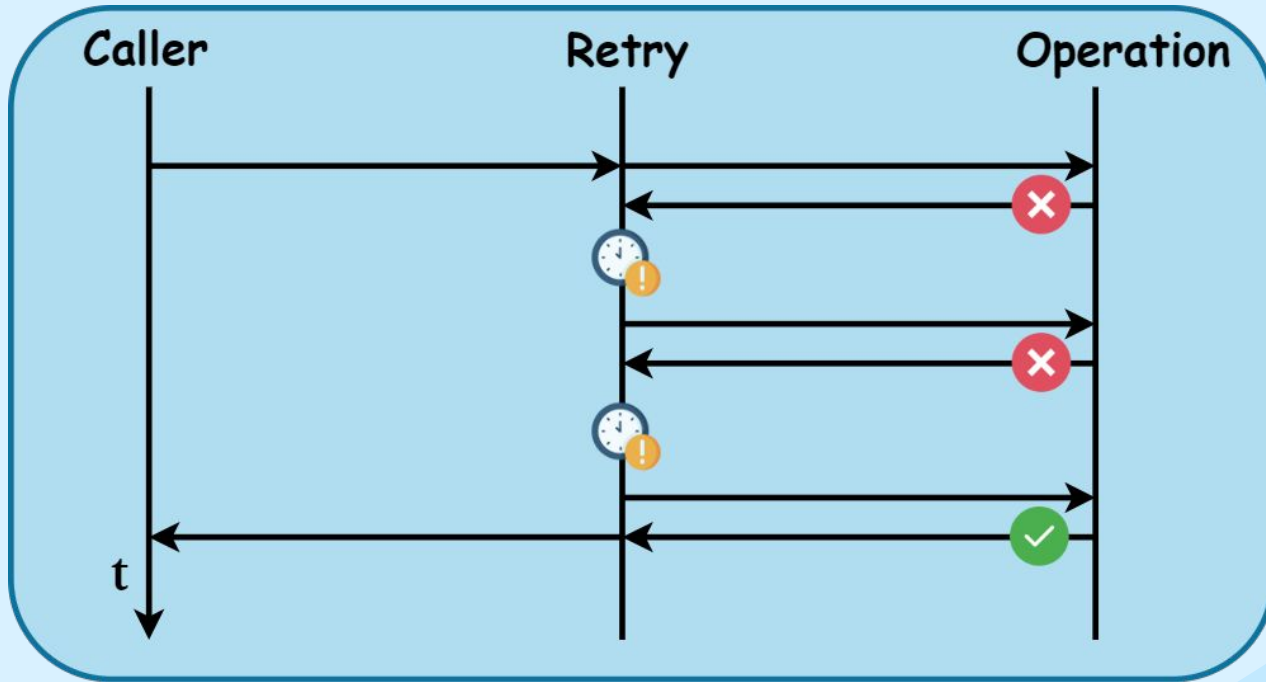


**Policies** - Define  
the mechanism  
behaviour

# Mechanism Model

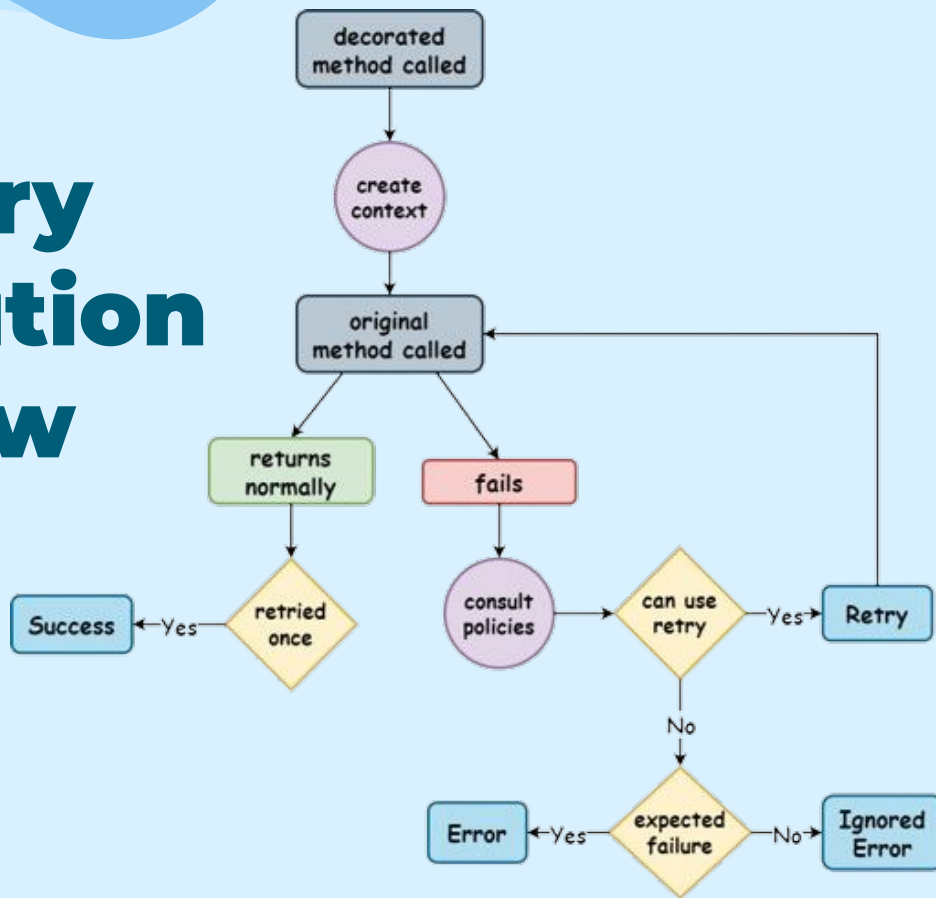


# Retry Mechanism

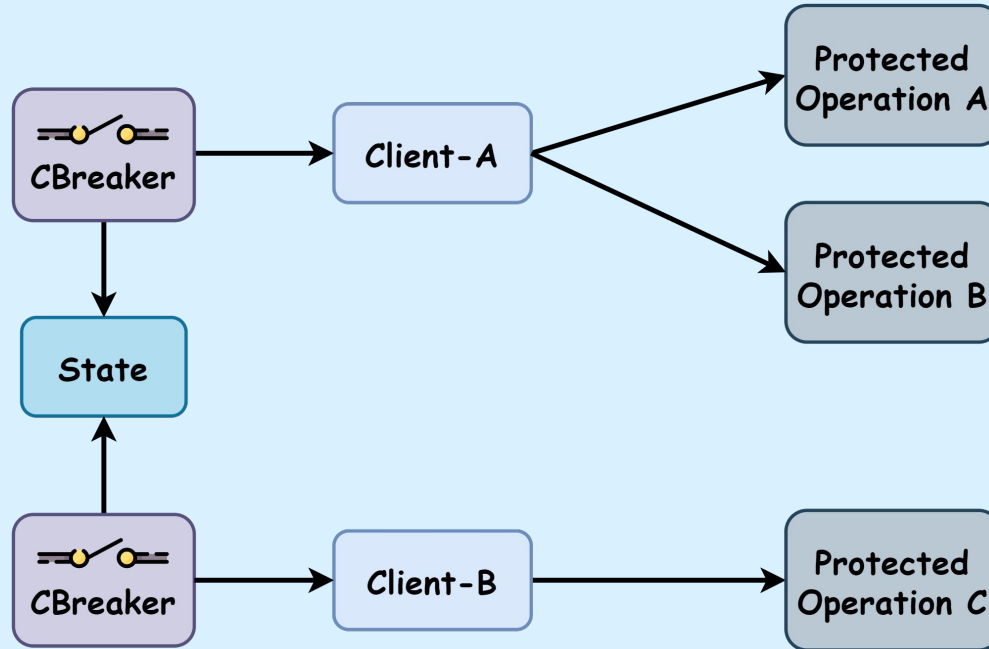




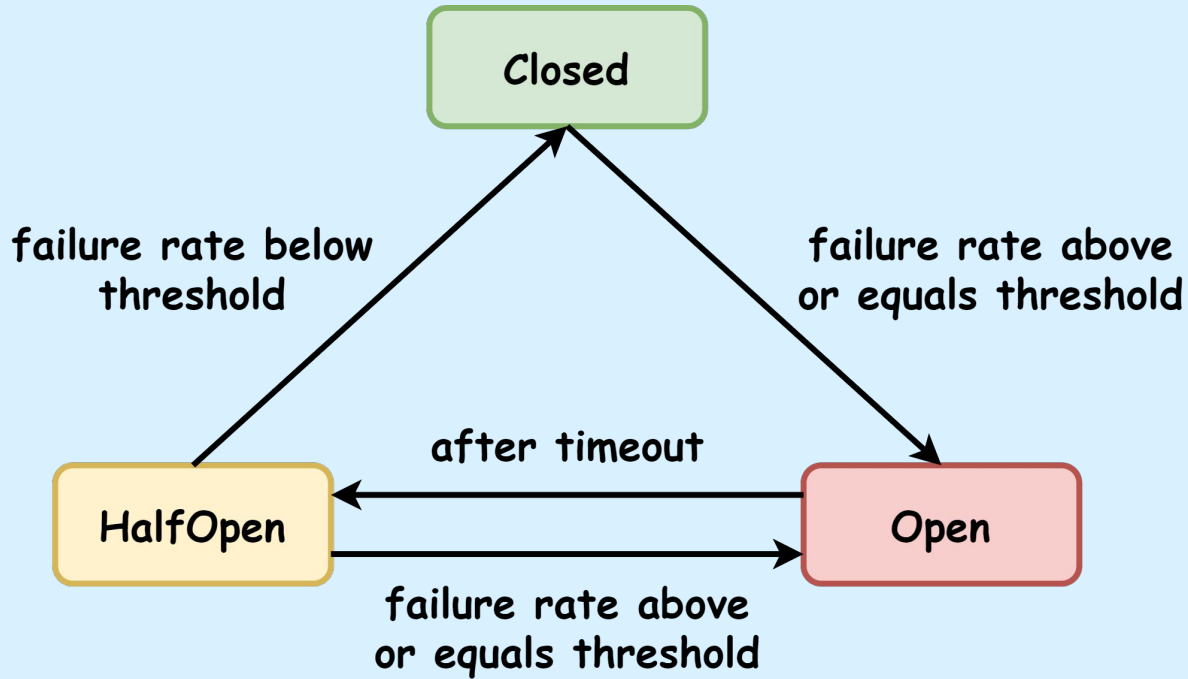
# Retry Execution Flow



# Circuit Breaker Mechanism



# Circuit Breaker States



# Circuit Breaker Example

