Kresil
Kotlin Resilience

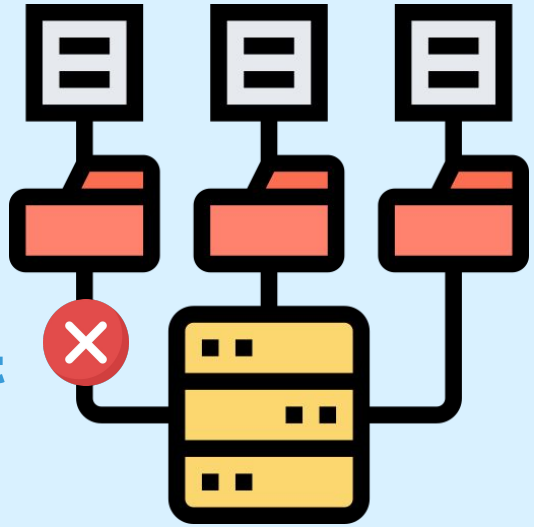
Kotlin Multiplatform library for fault-tolerance

ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

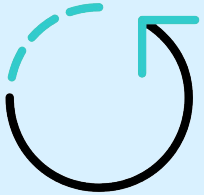
Supervisor: Prof. Pedro Félix
Author: Francisco Engenheiro - 49428

Resilience in Distributed Systems

If failure is inevitable, what can be done?

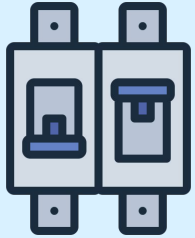


Mechanisms



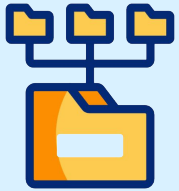
Retry - Repeats failed executions

Rate Limiter - Limits executions/period

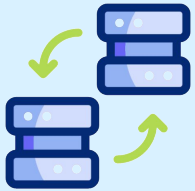


Circuit Breaker - Temporarily blocks possible failures

Time Limiter - Limits duration of execution



Cache - Memorizes a successful result



Fallback - Defines an action to fallback on failure

And many more...

Existing Solutions

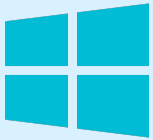
Resilience4j

Netflix's Hystrix

Kotlin Multiplatform

Kotlin/JS

Kotlin/Native

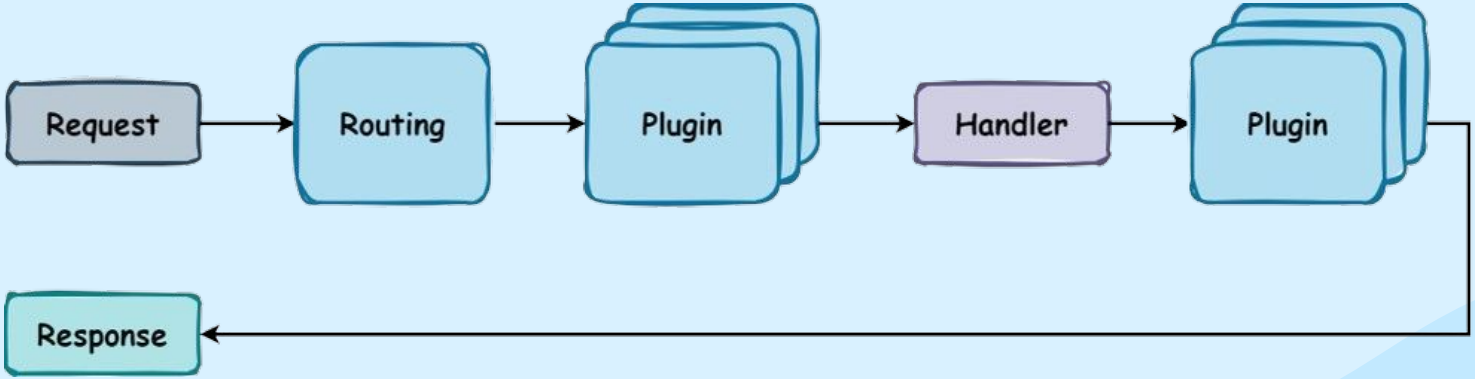


Kotlin/Android

Kotlin/JVM

Ktor Framework

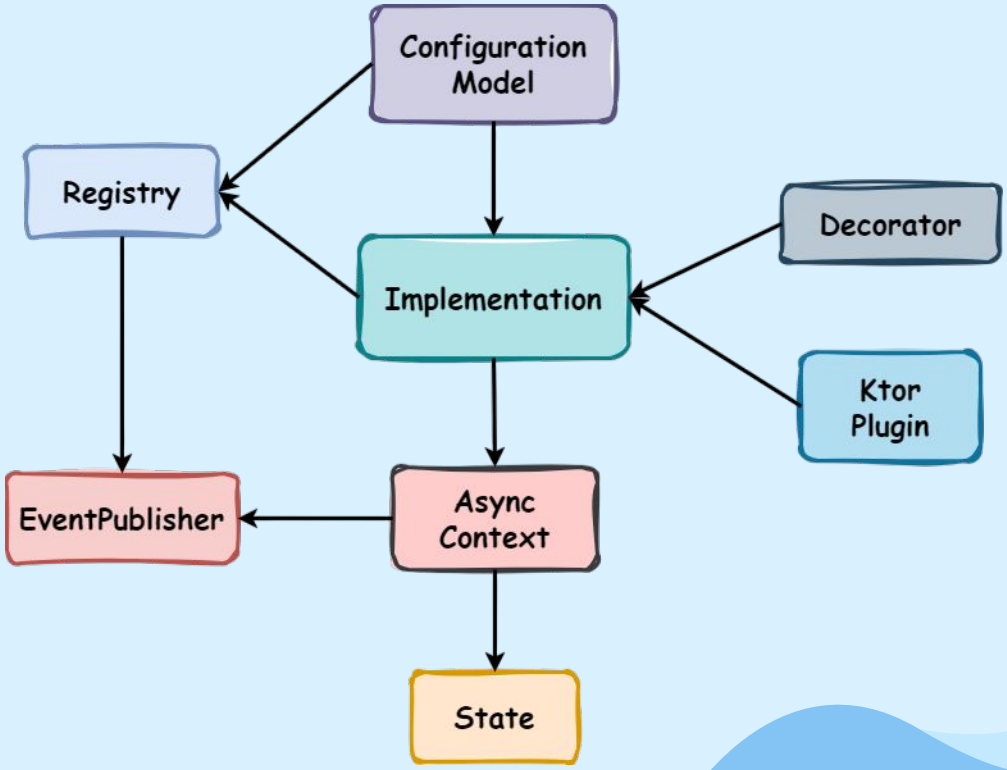
- Built with Kotlin Multiplatform;
- Enables asynchronous server and client development;
- Based on the coroutines system;
- Modular



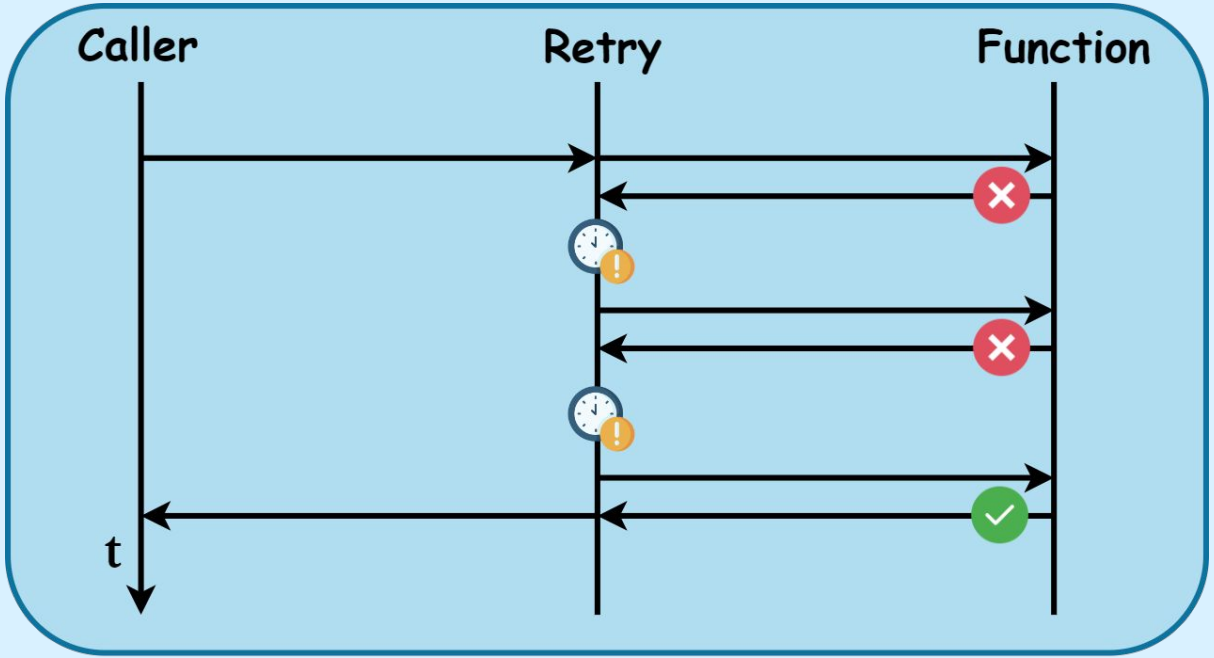
Mechanism Configuration

Policies - Define the mechanism behaviour

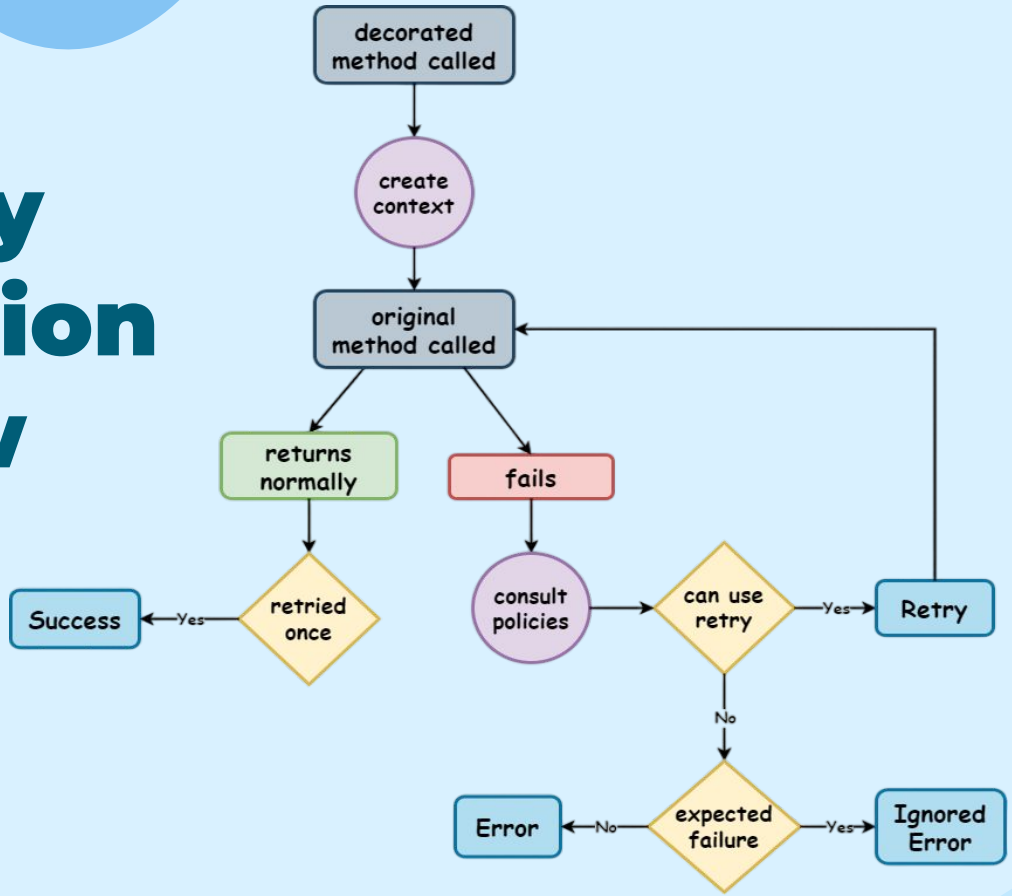
Mechanism Model



Retry Mechanism



Retry Execution Flow

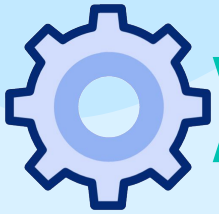


Code

```
val config: RetryConfig = retryConfig {
    maxAttempts( value: 3) // includes the first non-retry attempt
    retryIf { it is WebServiceException }
    delay(500.milliseconds)
}
val retry = Retry(config) // or Retry() for default config
```

```
retry.executeSuspendFunction {
    remoteService.suspendCall()
}
// or:
val decorated = retry.decorateSuspendFunction {
    remoteService.suspendCall()
}
// and call it later: decorated()
```

```
// for all events
retry.onEvent { println(it) }
retry.onRetry { currAttempt ->
    // handle retry event
}
retry.onError { error ->
    // handle error event
}
// and more...
```



Resilience4j | Ktor | Retry | Circuit Breaker | ?

- Retry Mechanism Study
- Tests to understand core functionality
- Design and Implementation Strategy for Kresil Mechanisms

- Client and Server Pipelines
- Custom Plugin API
- HttpRequest Retry Plugin

- Implementation
- Tests
- All targets support
- Ktor Plugin

- Implementation
- Tests
- All targets support
- Ktor Plugin