

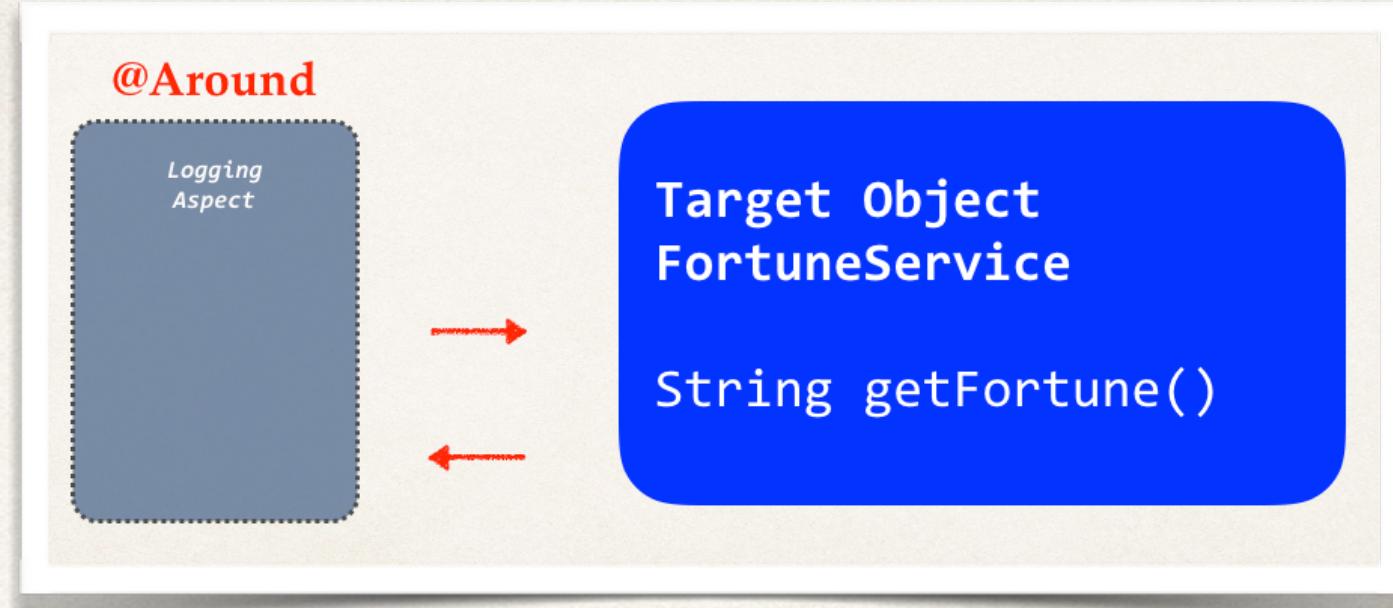
Aspect-Oriented Programming (AOP)

@Around Advice - Rethrow Exception

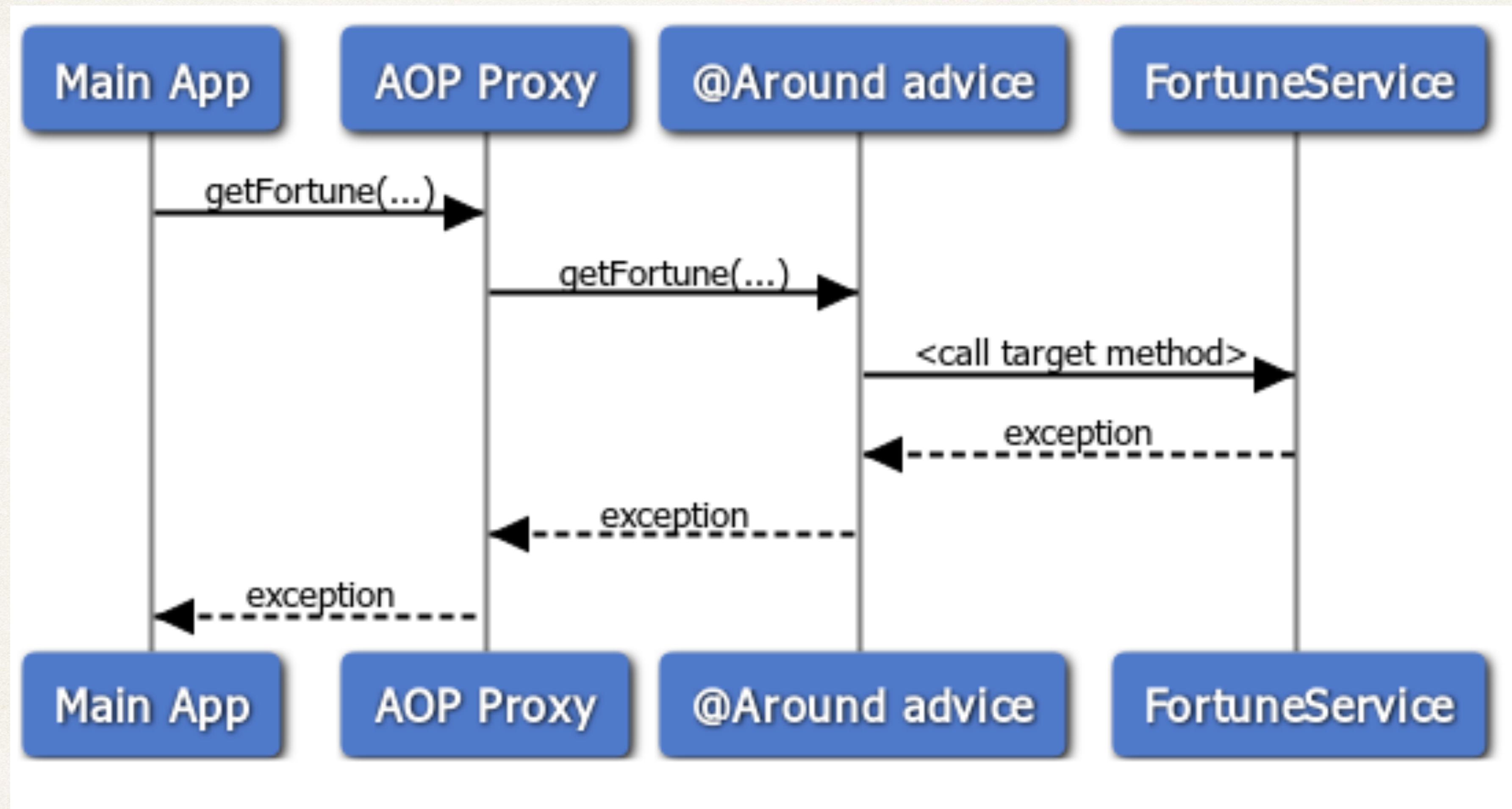


ProceedingJoinPoint - Revisited

- For an exception thrown from proceeding join point
 - You can handle / swallow /stop the exception
 - **Or you can simply rethrow the exception**



Sequence Diagram



Rethrow Exception

```
@Around("execution(* com.luv2code.aopdemo.service.*.getFortune(..))")
public Object afterGetFortune(
    ProceedingJoinPoint theProceedingJoinPoint) throws Throwable {

}

}
```

Rethrow Exception

```
@Around("execution(* com.luv2code.aopdemo.service.*.getFortune(..))")
public Object afterGetFortune(
    ProceedingJoinPoint theProceedingJoinPoint) throws Throwable {

    try {
        // let's execute the method
        Object result = theProceedingJoinPoint.proceed();

        return result;
    }

}
```

Rethrow Exception

```
@Around("execution(* com.luv2code.aopdemo.service.*.getFortune(..))")
public Object afterGetFortune(
    ProceedingJoinPoint theProceedingJoinPoint) throws Throwable {

    try {
        // let's execute the method
        Object result = theProceedingJoinPoint.proceed();

        return result;
    }
    catch (Exception exc) {
        // log exception
        System.out.println("@Around advice: We have a problem " + exc);

        // rethrow it
        throw exc;
    }
}
```

FAQ: JoinPoint vs ProceedingJoinPoint

- When to use JoinPoint vs ProceedingJoinPoint?
- Use JoinPoint with following advice types
 - @Before, @After, @AfterReturning, @AfterThrowing
- Use ProceedingJoinPoint with following advice type
 - @Around