

Aspect-Oriented Programming (AOP)

@AfterThrowing Advice

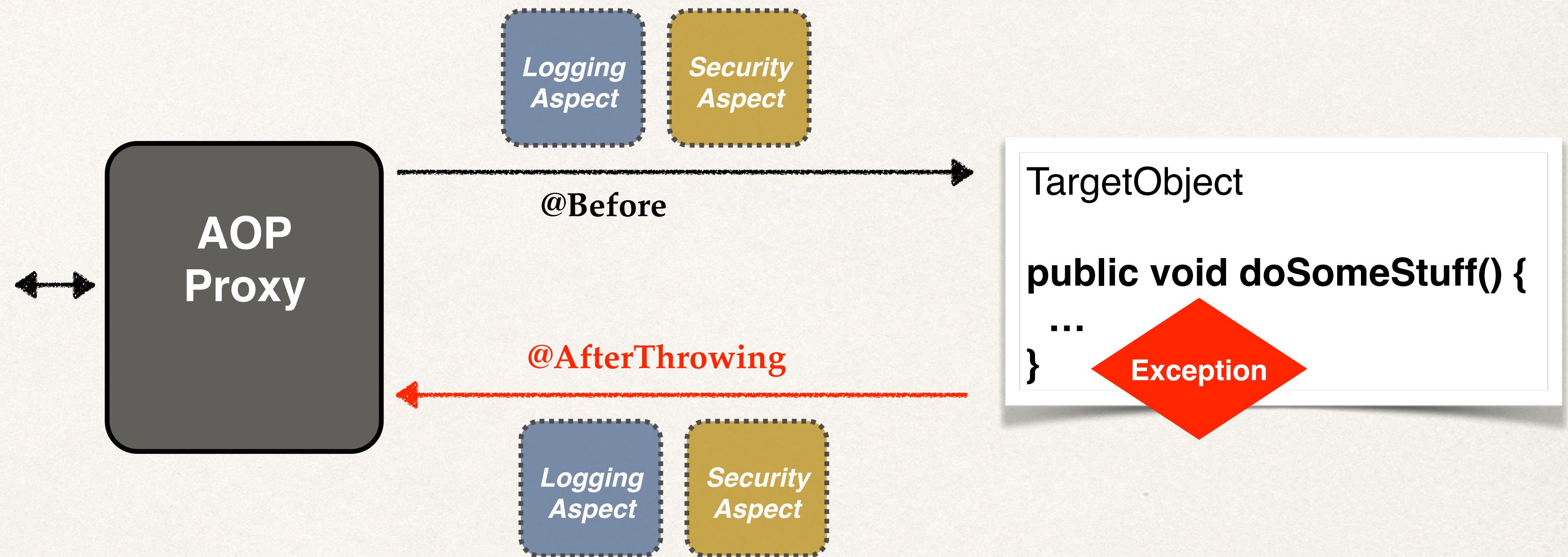


Advice Types

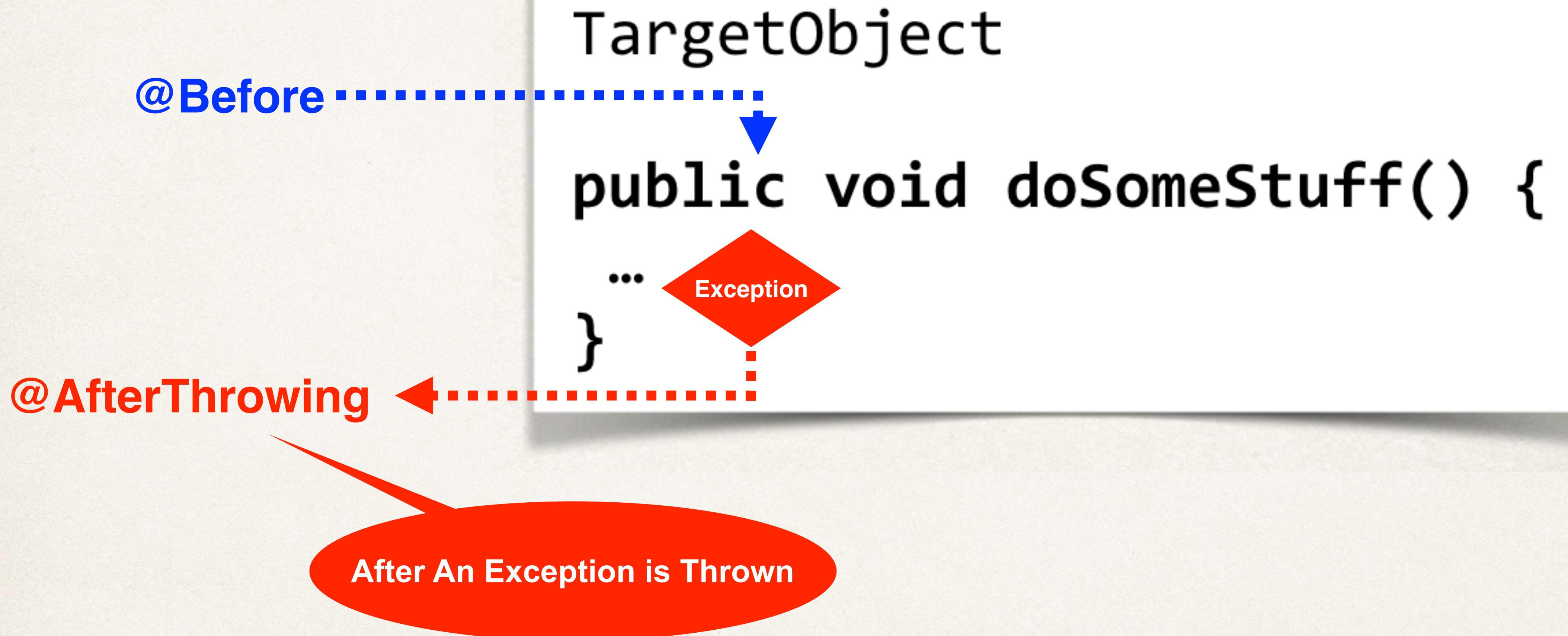
- **Before advice:** run before the method
- **After returning advice:** run after the method (success execution)
- **After throwing advice:** run after method (if exception thrown)
- **After finally advice:** run after the method (finally)
- **Around advice:** run before and after method

@AfterThrowing Advice - Interaction

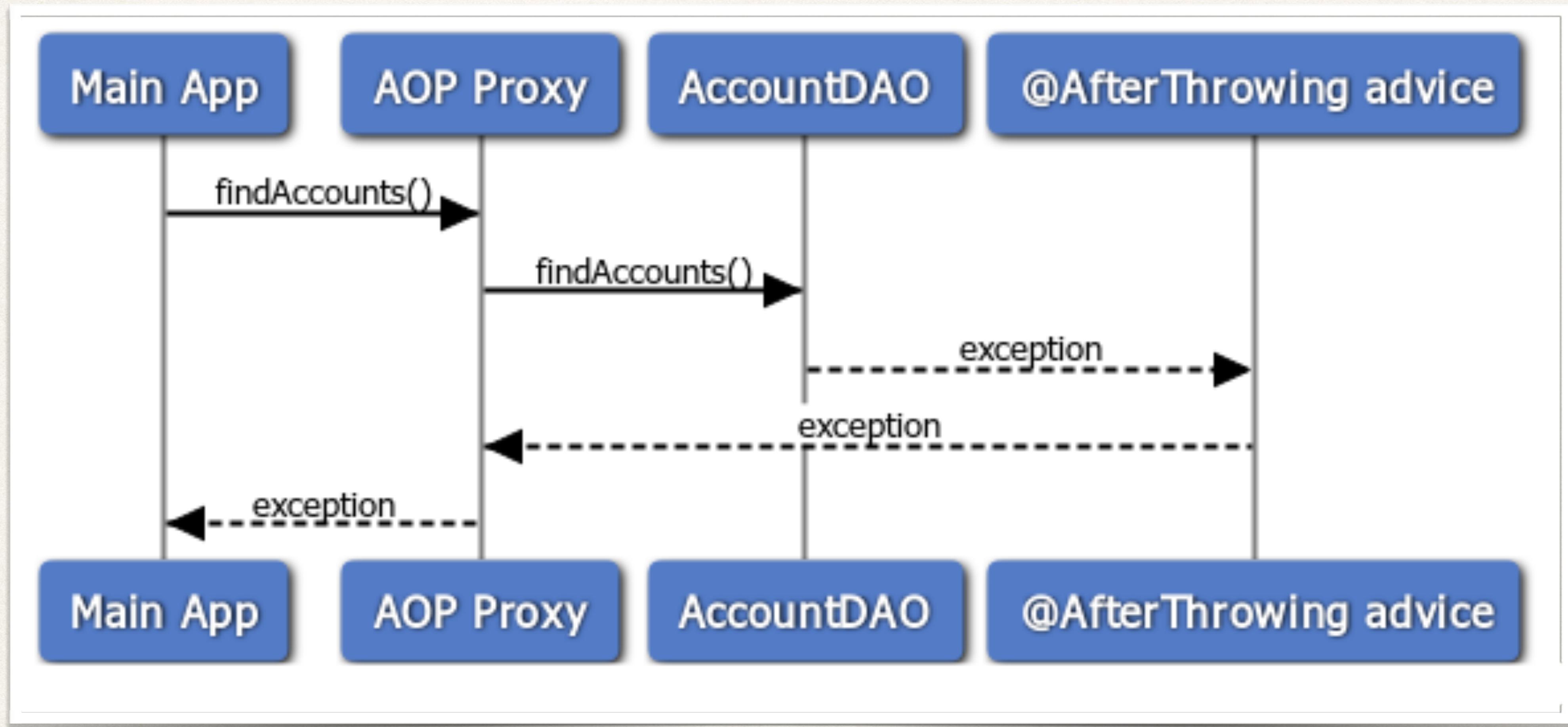
```
MainApp  
// call target object  
targetObj.doSomeStuff();
```



Advice - Interaction



Sequence Diagram



@AfterThrowing Advice - Use Cases

- Log the exception
- Perform auditing on the exception
- Notify DevOps team via email or SMS
- Encapsulate this functionality in AOP aspect for easy reuse

Example

- Create an advice that will run after an exception is thrown



@AfterThrowing Advice

- This advice will run after an exception is thrown

```
@AfterThrowing("execution(* com.luv2code.aopdemo.dao.AccountDAO.findAccounts(..))")
public void afterThrowingFindAccountsAdvice() {

    System.out.println("Executing @AfterThrowing advice");

}
```

Access the Exception object

- Need to access the exception object



Access the Exception

```
@AfterThrowing(  
    pointcut="execution(* com.luv2code.aopdemo.dao.AccountDAO.findAccounts(..))",  
    throwing="theExc")  
public void afterThrowingFindAccountsAdvice() {  
}  
}
```

Access the Exception

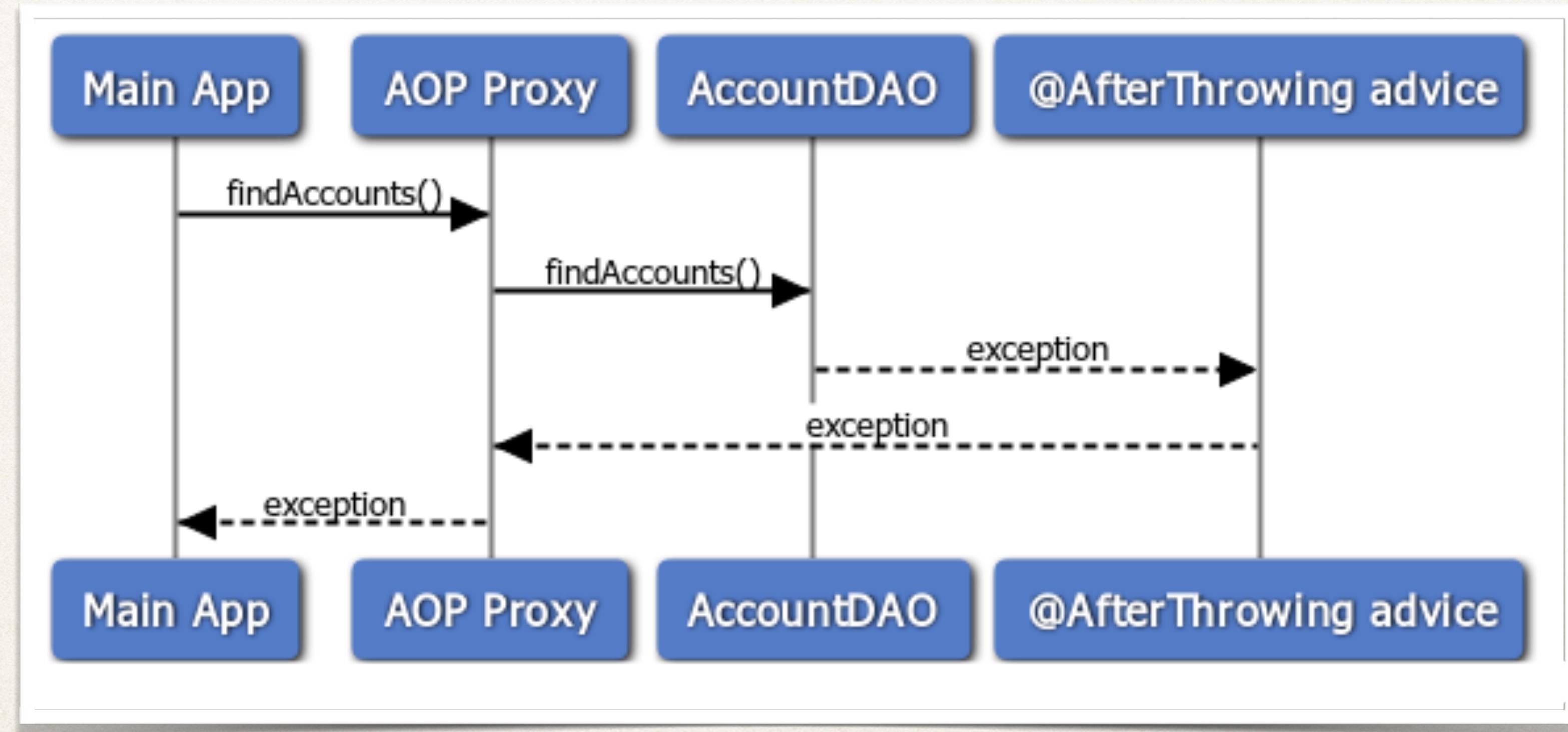
```
@AfterThrowing(  
    pointcut="execution(* com.luv2code.aopdemo.dao.AccountDAO.findAccounts(..))",  
    throwing="theExc")  
public void afterThrowingFindAccountsAdvice(  
    JoinPoint theJoinPoint, Throwable theExc) {  
  
}
```

Access the Exception

```
@AfterThrowing(  
    pointcut="execution(* com.luv2code.aopdemo.dao.AccountDAO.findAccounts(..))",  
    throwing="theExc")  
public void afterThrowingFindAccountsAdvice(  
    JoinPoint theJoinPoint, Throwable theExc) {  
  
    // log the exception  
    System.out.println("\n=====>>> The exception is: " + theExc);  
  
}
```

Exception Propagation

- At this point, we are only intercepting the exception (reading it)
- However, the exception is still propagated to calling program



Exception Propagation

- If you want to stop the exception propagation
 - then use the **@Around** advice
 - covered in later videos