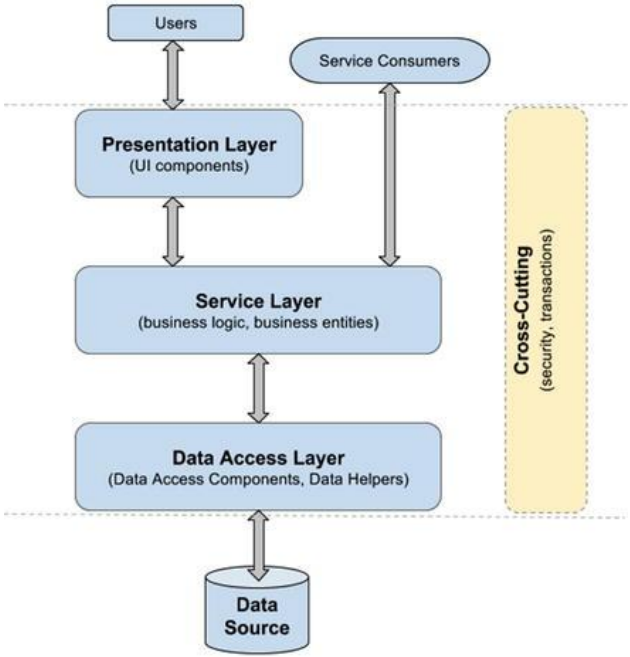
Spring Day 04 - AOP

@Pointcut
@Aspect
Transactions
AOP
Weaving
Security
Join-Point
Advice
Spring-AOP

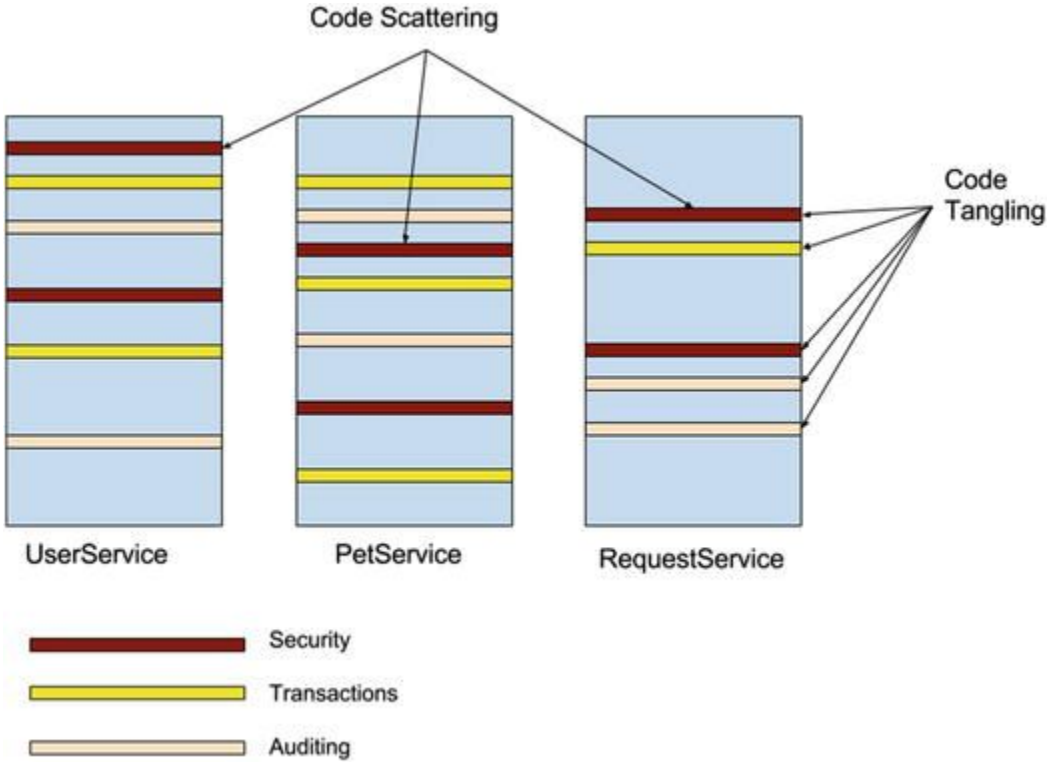
AOP Concepts

- AOP is a type of programming that aims to help with separation of cross-cutting concerns to increase modularity; it implies declaring an aspect class that will alter the behavior of base code, by applying advices to specific join points, specified by pointcuts.

Cross-Cutting concerns



Problems Solved by AOP



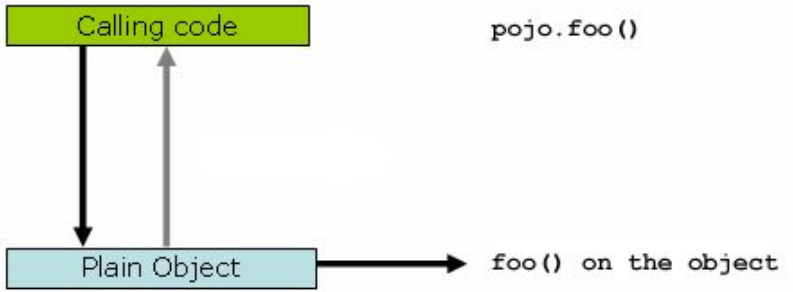
AOP Terminology

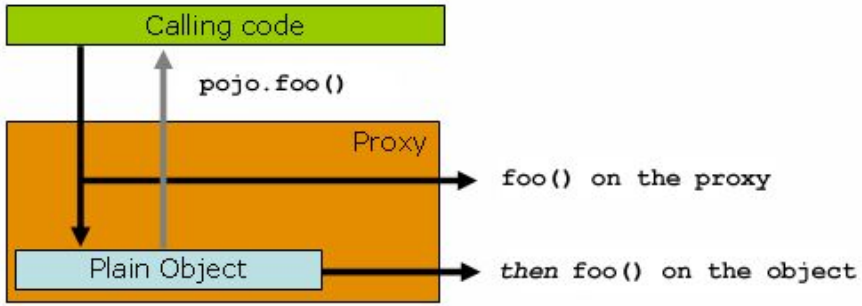
- **Aspect**: a class containing code specific to a cross-cutting concern
- **Join point**: all possible points where an Aspect can be applied
- **Pointcut**: a predicate used to identify join points. Advice definitions are associated with a pointcut expression and the advice will execute on any join point matching the pointcut expression.
- **Advice**: action taken by an aspect at a join point
- **Weaving**: combining advised object with the advice code

Spring AOP

- Example
- Spring AOP Documentation

Spring AOP


Calling code
pojo.foo()
Plain Object
foo() on the object


Calling code
pojo.foo()
Proxy
foo() on the proxy
Plain Object
then foo() on the object

Summary and Best Practices

- Spring AOP advices only public Join Points
- Aspects can be applied only to Spring Beans
- Avoid whenever possible
- Advise only classes that implement an interface
- Advise only methods exposed through an interface
- Call Advised bean only through interface

@Pointcut
@Aspect
Transactions
AOP
Weaving
Security
Join-Point
Advice
Spring-AOP