

AOP in Objective C

Aspektororientierte Programmierung

Agenda

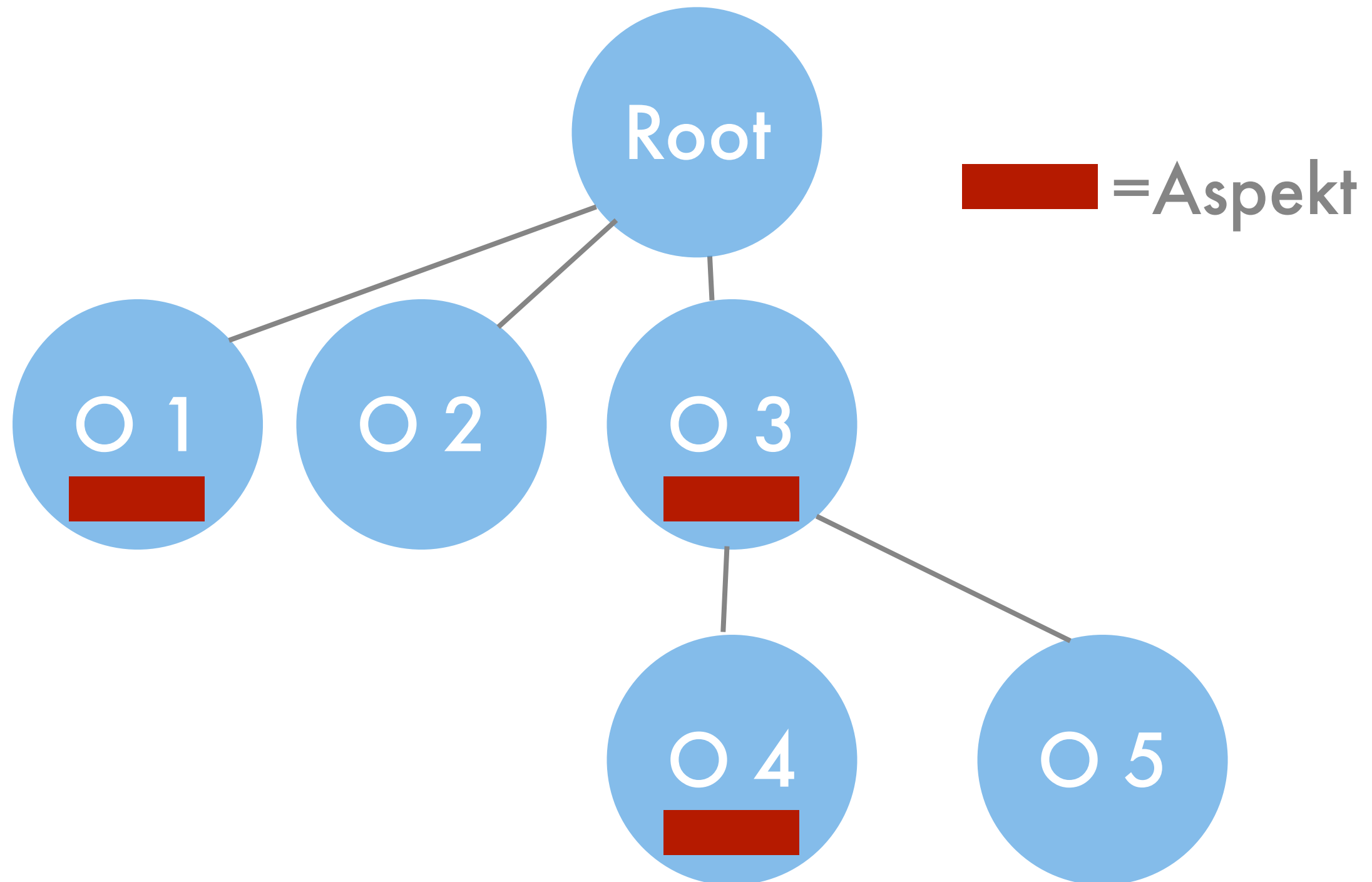
- Was ist AOP?
- Umsetzung AOP in Objective-C
- Bestehende Libraries

Concerns & Aspekte

Anwendungslogik
Core Concerns

Aspekte
Cross-Cutting Concerns

Cross-Cutting



Die 'tweet' Methode

```
- (BOOL)tweet:(NSString*)msg{
    if (!loggedIn) {
        return NO;
    }

    NSString* string = [NSString stringWithFormat:@"http://twitter.com/tweet?%@", msg];
    NSURL* url = [NSURL URLWithString:string];
    NSError* error = nil;
    NSString* response = [NSString stringWithContentsOfURL:url
                                                         encoding:NSUTF8StringEncoding error:&error];

    if (response && !error) {
        NSLog(@"Tweeting message %@", msg);
        return YES;
    }
    return NO;
}
```

Die 'tweet' Methode

```
- (BOOL)tweet:(NSString*)msg{
```

```
    if (!loggedIn) {  
        return NO;  
    }
```

Aspekte



```
    NSString* string = [NSString stringWithFormat:@"http://twitter.com/tweet?%@", msg];
```

```
    NSURL* url = [NSURL URLWithString:string];
```

```
    NSError* error = nil;
```

```
    NSString* response = [NSString stringWithContentsOfURL:url  
                        encoding:NSUTF8StringEncoding error:&error];
```

```
    if (response && !error) {
```

```
        NSLog(@"Tweeting message %@", msg);  
        return YES;
```

```
    }
```

```
    return NO;
```

```
}
```

Aspects & Advices

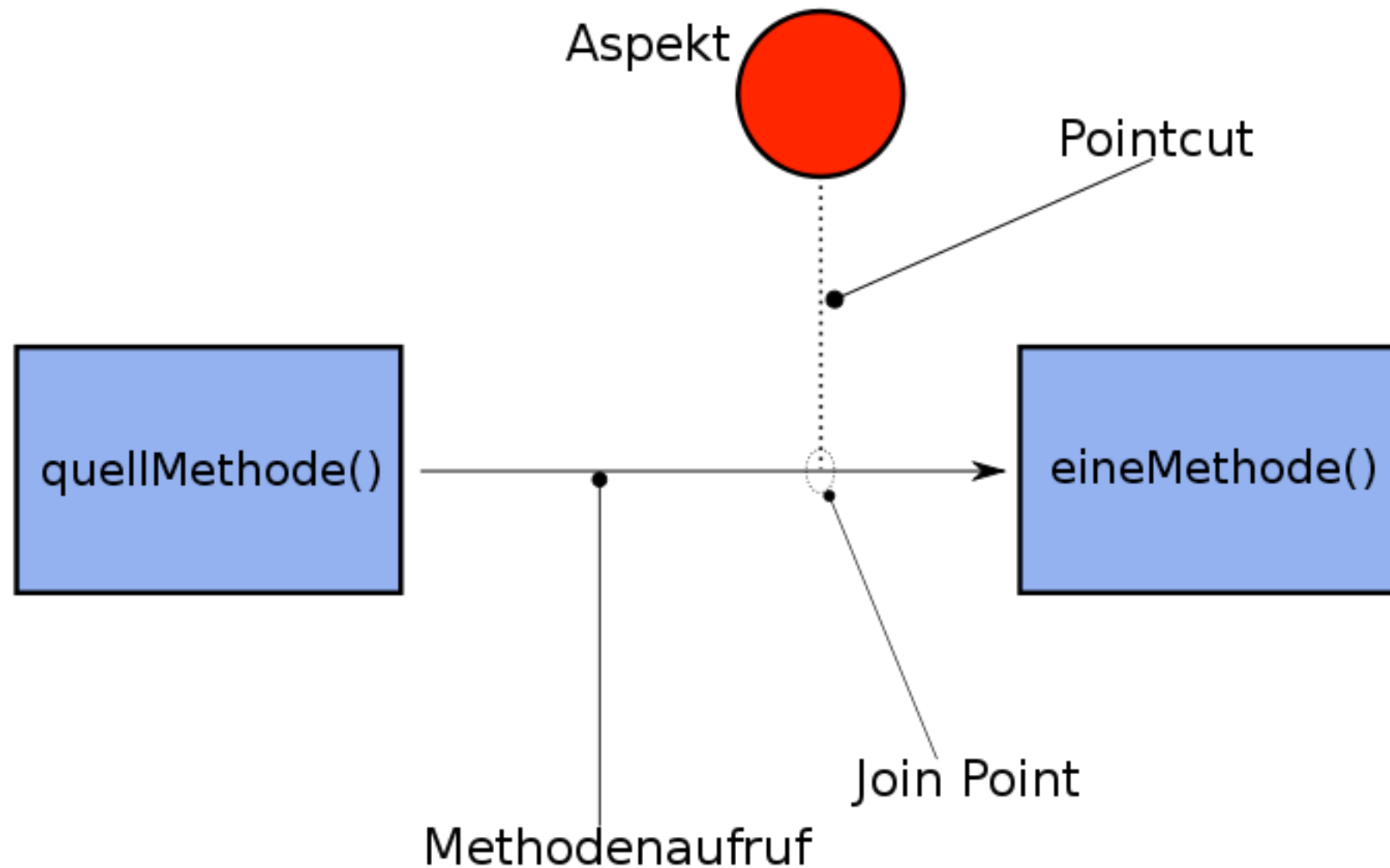


Aspect: Check Login / Advice Before

Methode: Tweet

Aspect: Log to Console / Advice After

Join Points & Pointcuts



AOP API

- Aspekte definieren
- Aspekten Advices zuordnen
- Join Points definieren
- Join Points Aspekten zuordnen

Beispiel

```
Aspect loggingAspect =  
    ^(NSInvocation* invocation, NSDictionary* data){  
        NSLog(@"Method finished excecution");  
        return (id)nil;  
    };
```

```
[aopManager insertAspect:loggingAspect  
    afterMethod:@selector(sayHello)  
    ofClass:[AOPTestClass class]];
```

Beispiel

```
Aspect loggingAspect =  
    ^(NSInvocation* invocation, NSDictionary* data){  
        NSLog(@"Method finished excecution");  
        return (id)nil;  
    };  
;
```

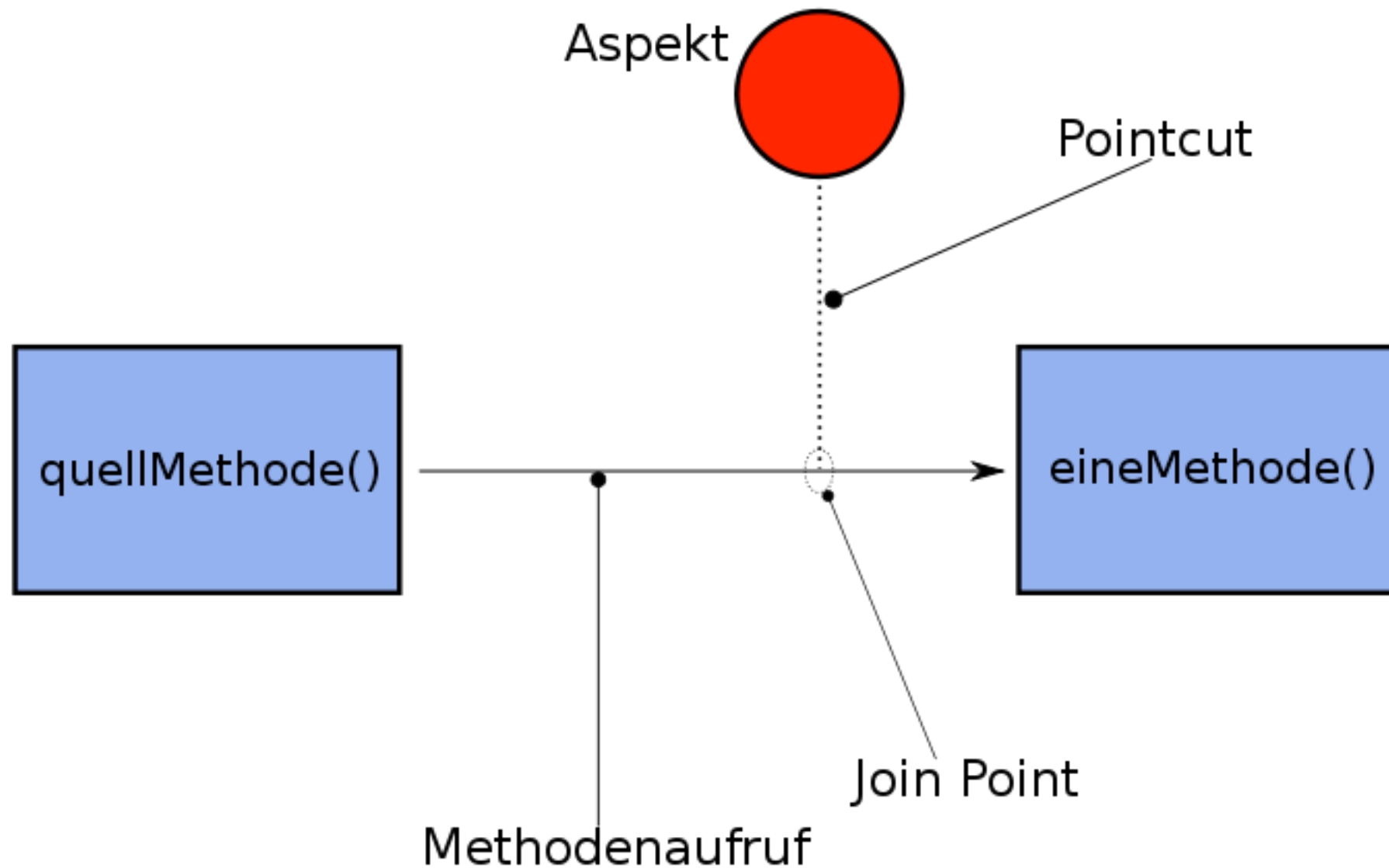
Aspekt

```
[aopManager insertAspect:loggingAspect  
    afterMethod:@selector(sayHello)  
    ofClass:[AOPTestClass class]];
```

Advice

Join Point

Umsetzung?



Aspekte Ausführen

Proxies

Method
Swizzling

isa-
Zeiger
ändern

Proxies

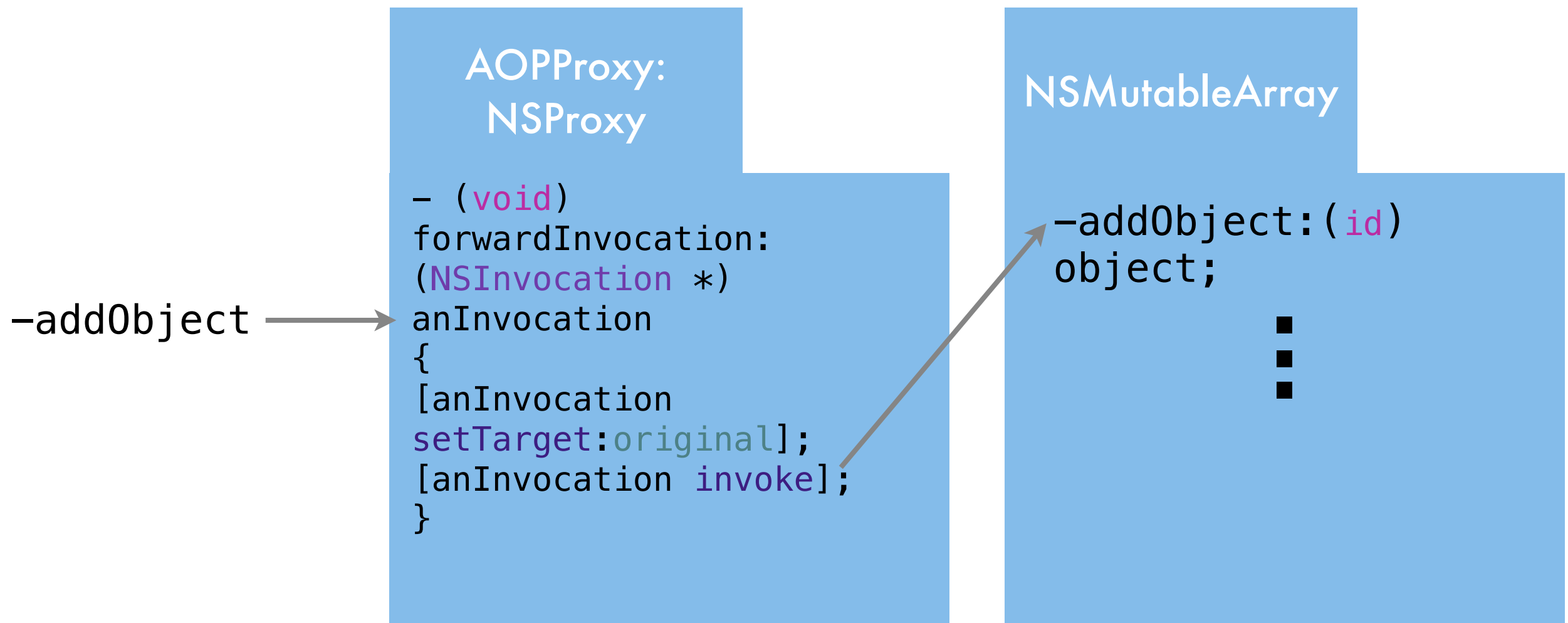
Proxies (1)

```
NSMutableArray* testArray = [[A0PProxy alloc]
initWithNewInstanceOfClass:[NSMutableArray class]];
```

```
[(A0PProxy*)testArray
    interceptMethodStartForSelector:@selector(addObject:)
        withInterceptorTarget:self
        interceptorSelector:@selector(addInterceptor:)];
```

```
- (void) addInterceptor:(NSInvocation *) i {
    NSLog(@"ADD intercepted.");
}
```

Proxies (2)



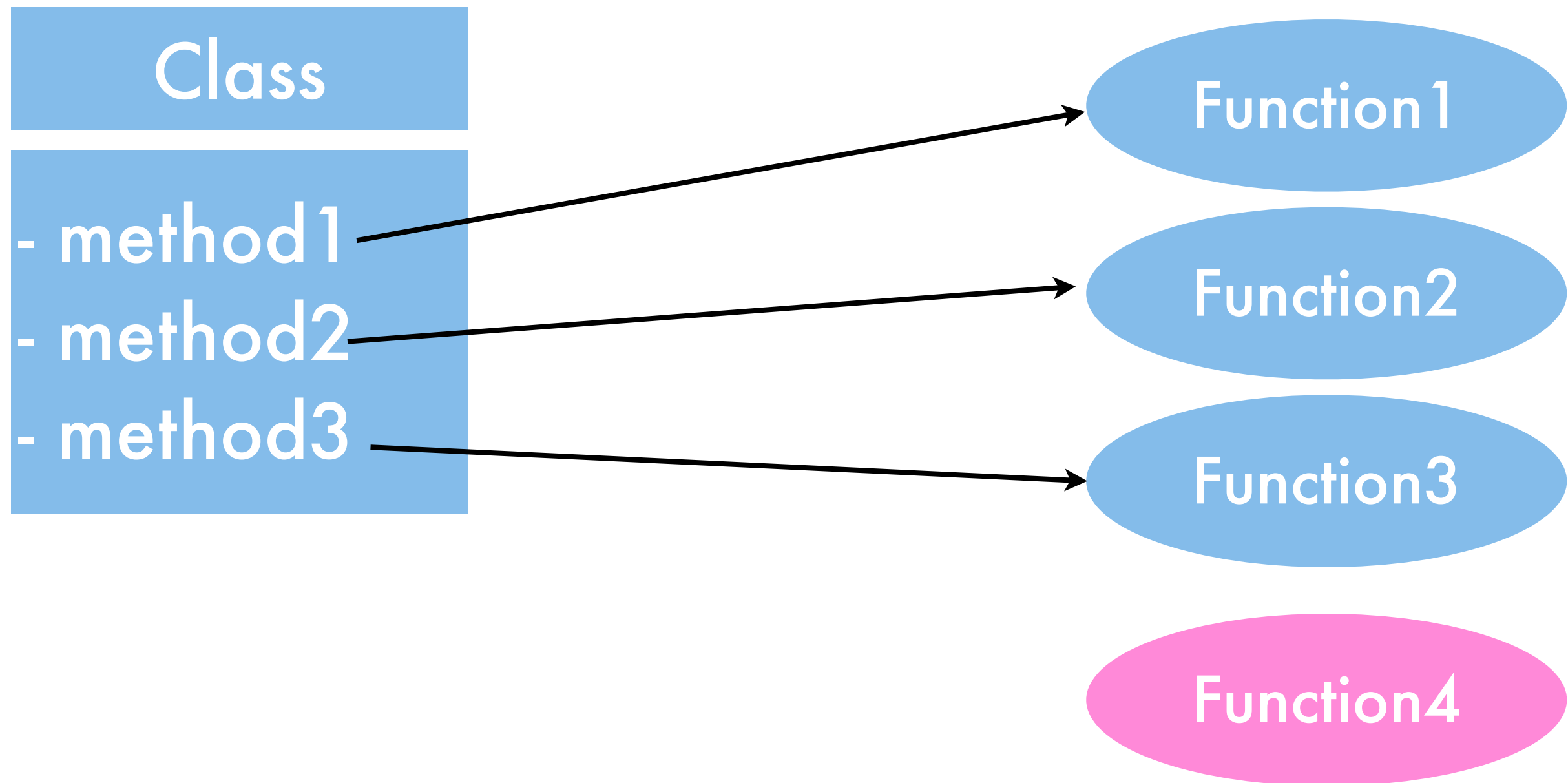
DEMO Proxy

Vorteile Proxy-Ansatz

- High-Level
- DI-Libraries
- Einfachere Implementierung
- Einfaches Unittesting

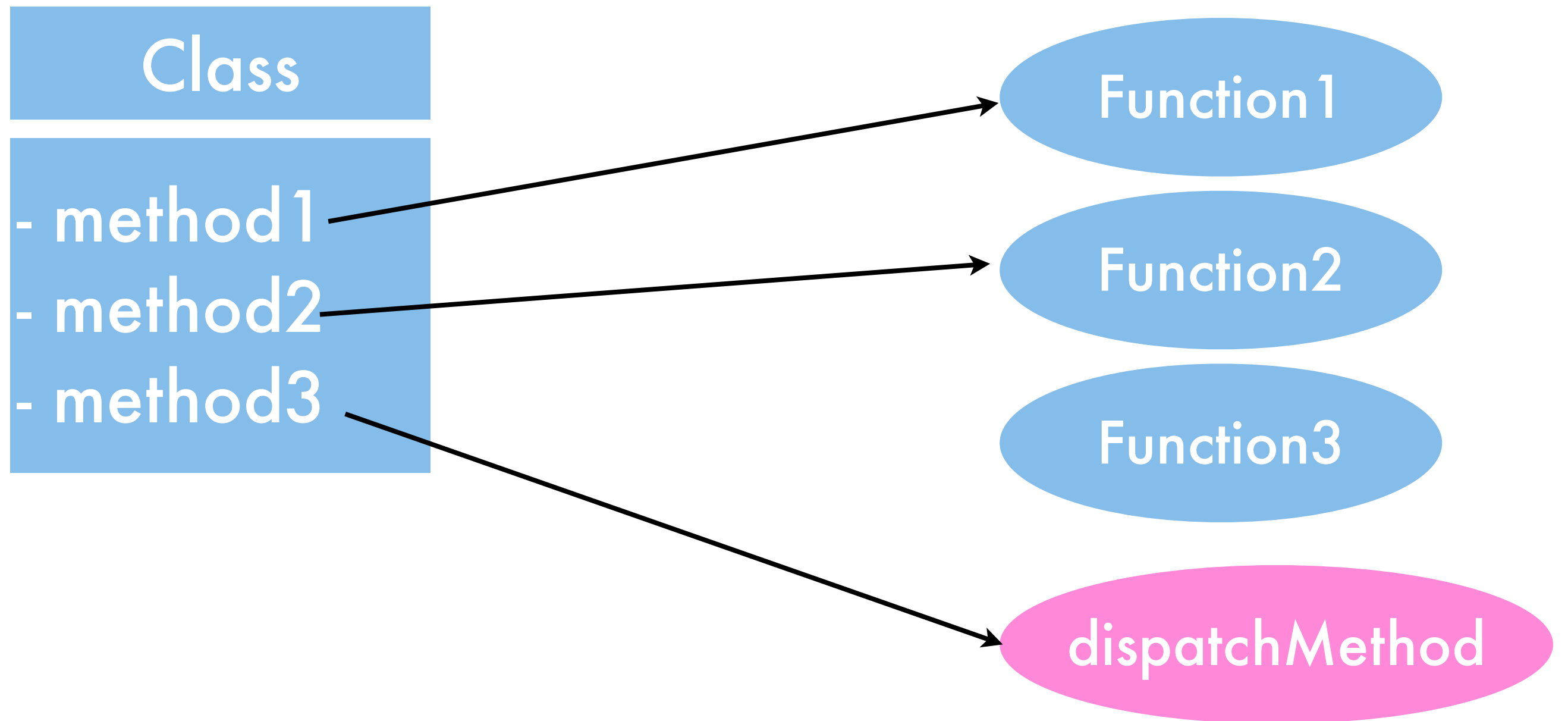
Method Swizzling

Method Swizzling



`class_replaceMethod, method_setImplementation`

Method Swizzling (2)



`class_replaceMethod, method_setImplementation`

Method Swizzling (1)

NSMutableArray

```
-addObject:(id)  
object;
```

dispatchMethod()

Method Swizzling (1)

NSMutableArray

dispatchMethod()

dispatchMethod



Iteration über alle before-Aspekte

Originalmethode ausführen

Iteration über alle after-Aspekte

DEMO Method Swizzling

Vorteile Method Swizzling

- Einfaches einbauen in bestehenden Code
- Performanz?

DEMO ISA

Frameworks

AspectCocoa

- Download unauffindbar
- Aboandonware
- [www.cocoadev.com/index.pl?
AspectCocoa](http://www.cocoadev.com/index.pl?AspectCocoa)

MGAOP

- 2847 loc
- Nicht dokumentiert
- Fast nicht kommentiert
- Method swizzling unsauber
- Kompliziert zu benutzen
- Class Clusters
- github.com/mgebele/MGAOP

AOP-In-Objective-C

- 395 loc
- Proxy-Basiert
- Gut dokumentiert und kommentiert
- Einfach zu benutzen
- github.com/moszi/AOP-in-Objective-C