AOP & Security

- Using Aspect Oriented Programming to Prevent Application Attacks

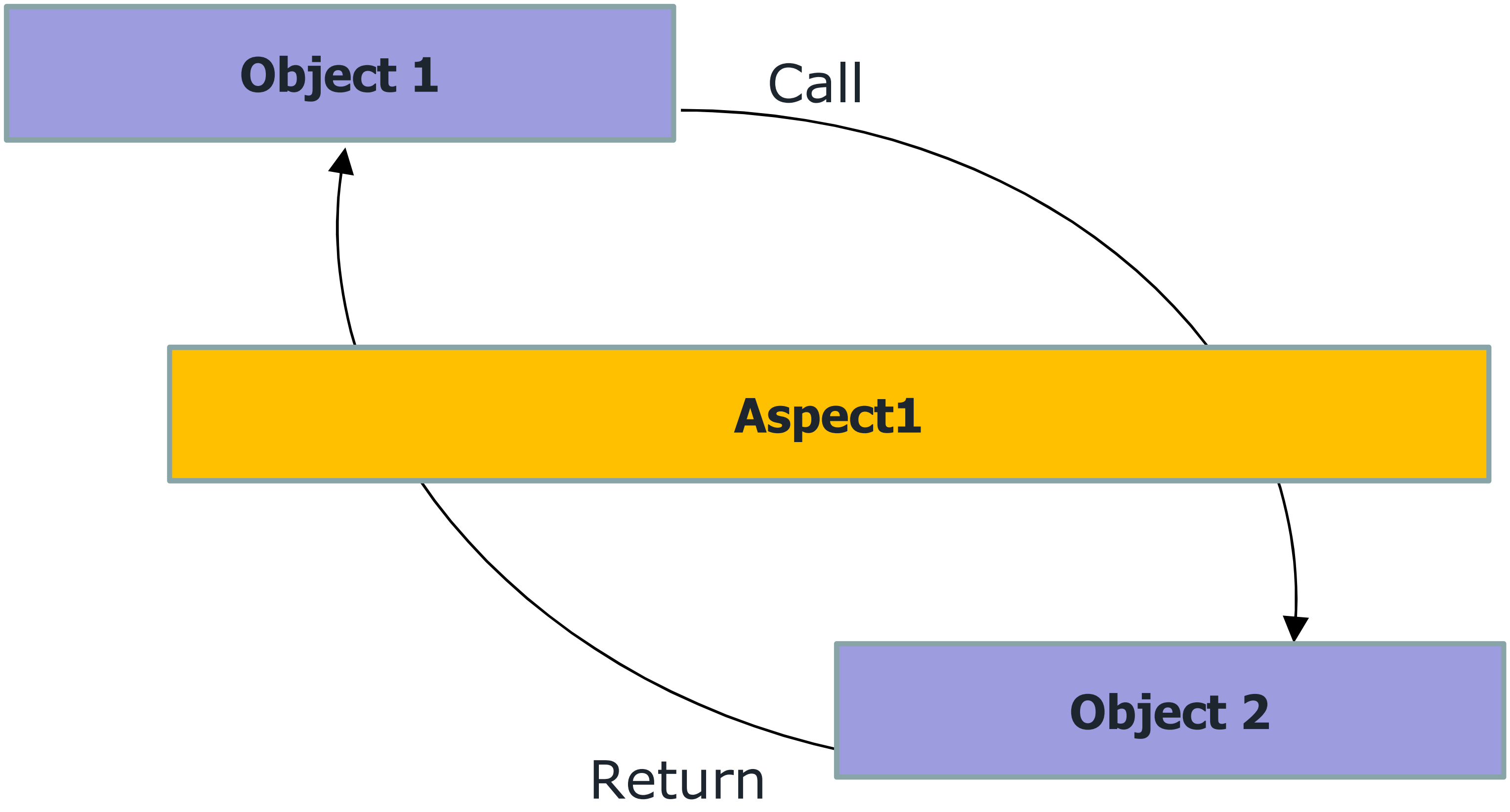
Nish Bhalla and Rohit Sethi, Security Compass

What is AOP?

```
public void throw (Ball b){
    otherPlayer.catch(b);
}
```

What is AOP?

No knowledge of aspect



Object 1

Call

Aspect1

Object 2

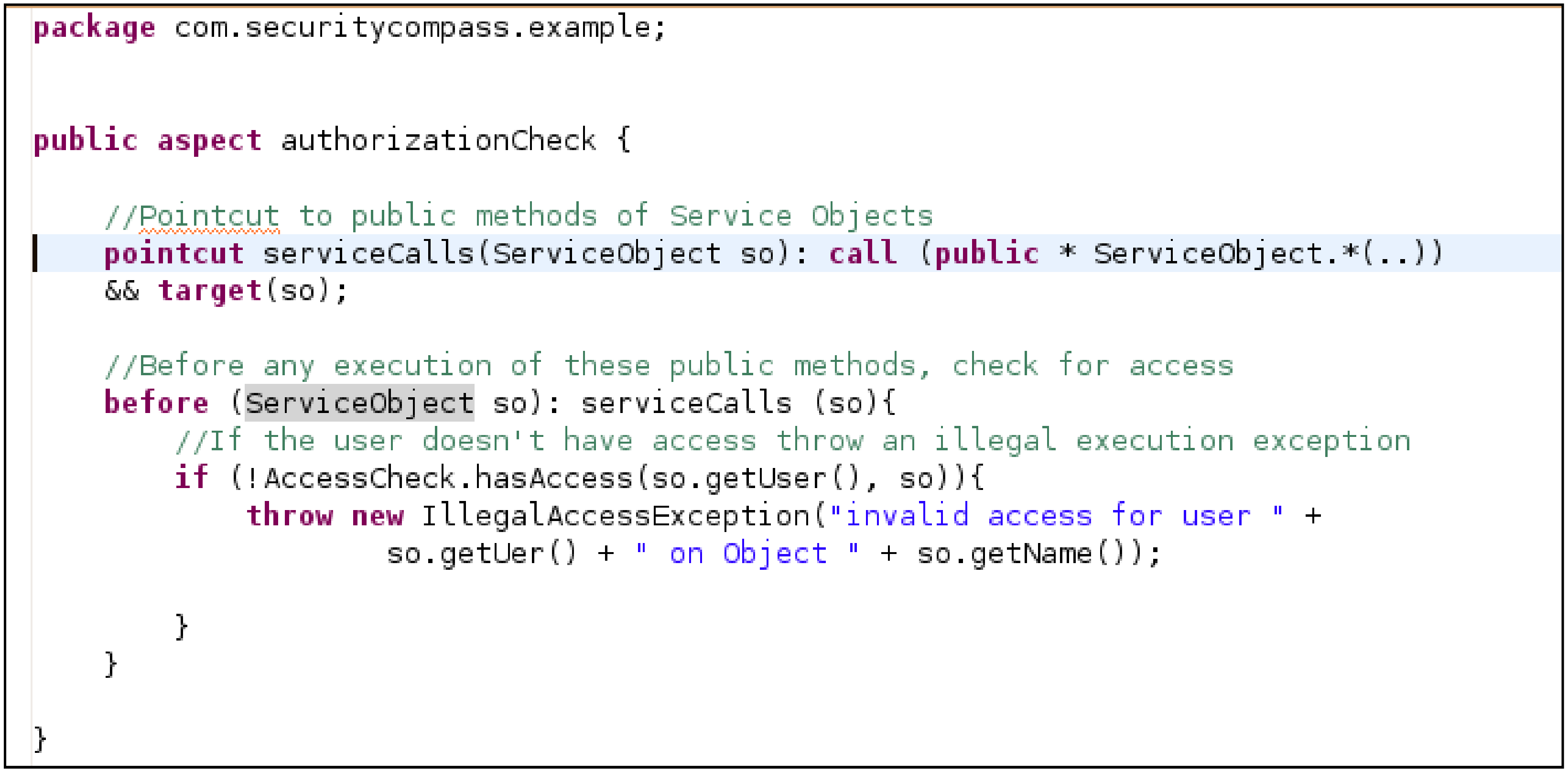
Return

No knowledge of aspect

AOP Daffodil video

Other Applications

- Example Authorization Check



```
package com.securitycompass.example;


public aspect authorizationCheck {

    //Pointcut to public methods of Service Objects
    pointcut serviceCalls(ServiceObject so): call (public * ServiceObject.*(..))
    && target(so);

    //Before any execution of these public methods, check for access
    before (ServiceObject so): serviceCalls (so){
        //If the user doesn't have access throw an illegal execution exception
        if (!AccessCheck.hasAccess(so.getUser(), so)){
            throw new IllegalAccessException("invalid access for user " +
                    so.getUer() + " on Object " + so.getName());

        }
    }

}
```

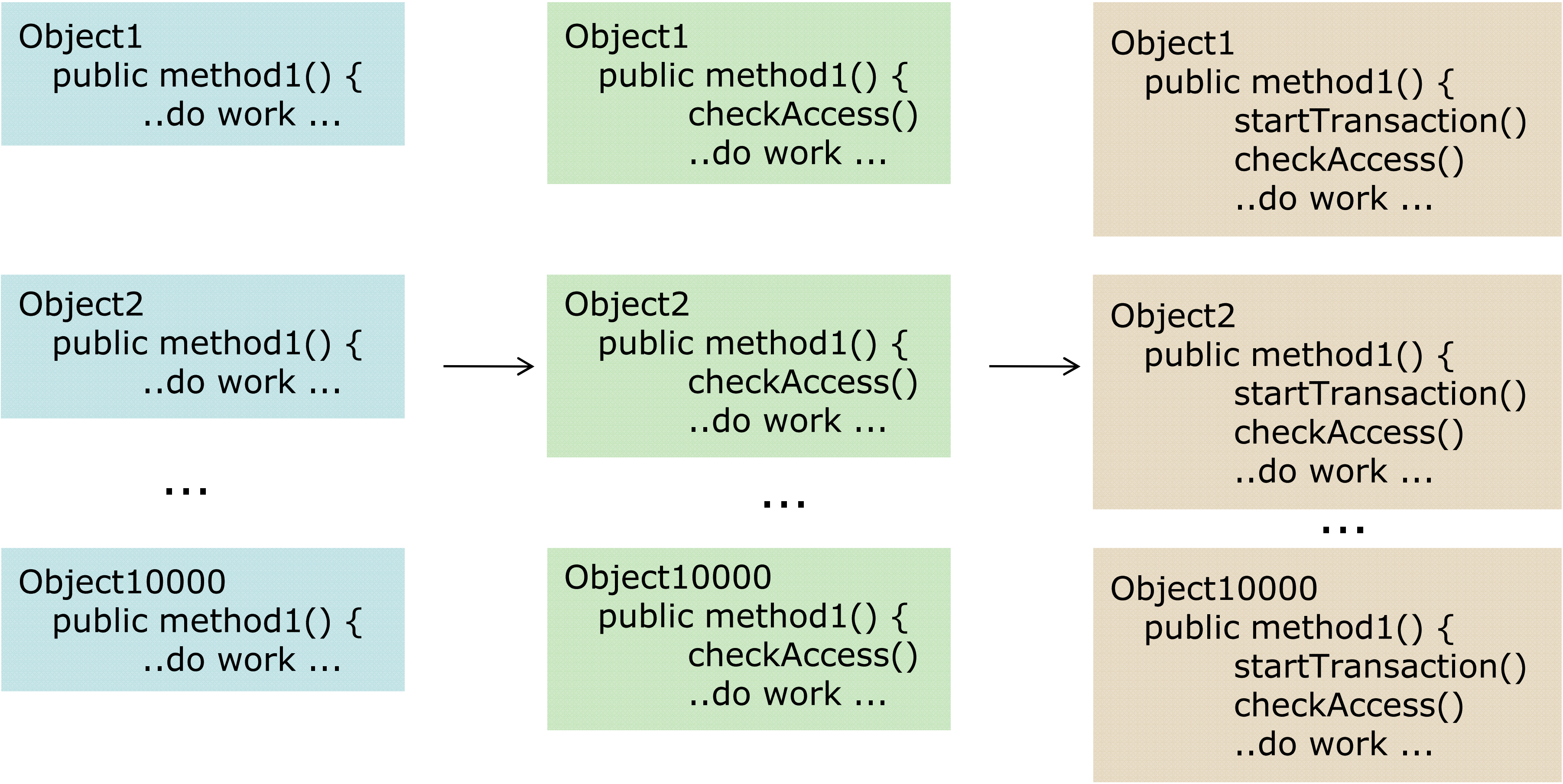
Other Applications

- Question to yourself:
  - Where else can you see AOP being used for security?

- We'll revisit this

Common Objections

- “We can achieve all of this in OOP”



```
Object1
   public method1() {
           ..do work ...

Object2
   public method1() {
           ..do work ...

...

Object10000
   public method1() {
           ..do work ...
```

```
Object1
   public method1() {
           checkAccess()
           ..do work ...

Object2
   public method1() {
           checkAccess()
           ..do work ...

...

Object10000
   public method1() {
           checkAccess()
           ..do work ...
```

```
Object1
   public method1() {
           startTransaction()
           checkAccess()
           ..do work ...

Object2
   public method1() {
           startTransaction()
           checkAccess()
           ..do work ...

...

Object10000
   public method1() {
           startTransaction()
           checkAccess()
           ..do work ...
```

Common Objections

- "AOP is unproven and unstable"
  - AspectJ 11 years old
  - Official Eclipse Project
  - Available in JBoss & Spring
  - Used internally in WebSphere

Common Objections

- “AOP is unproven and unstable”
  - IBM “[AOP] is vital for our survial”
  - Microsoft researching AOP for .Net
  - Industry: Siemens, Hitachi, SAP, Motorla

Common Objections

- “Performance overhead”
  - Legitimate concern for ‘runtime weaving’
  - ‘Compile-time weaving’ is very fast

Other Objections?

???

SQLi video

AOP Implementations

- AspectJ – Most popular
- Spring AOP & JBOSS AOP
- AOP.NET
- AspectC++
- PHPAspect

Adoption Obstacles

- Obstacles:
  - Lack of skill-set
  - Untested technology in your org
  - Changes to dev methodology
- Solutions:
  - Begin in lab
  - Incremental production implementation
  - Start with non-critical functions

Questions?

???