

Managing Exceptions Through Aspect Oriented Programming

INTRODUCTION



Sekhar Srinivasan

@sekharonline4u

www.sekhartheguru.net



Overview

**What is Aspect
Oriented
Programming**

AOP Terminology

**Configure
Application to
support AOP**

**Adding Logging
to Application**

**Manage
Exceptions using
AOP**



Understanding AOP



```
@Repository
public class AccountDAOImpl
    implements
AccountDAO{
@Override
public boolean saveAccount(Account
    Add Logging Code Here
...
}
@Override
public Account getAccount(Integer
acno) {
...
}
```



Develop
er



Manage
r

Add
Logging
functionalit
y to
saveAccoun
t method



```
@Repository
public class AccountDAOImpl
    implements
AccountDAO{
@Override
public boolean saveAccount(Account
    Add Logging Code Here

...
}

@Override
    Add Logging Code Here
acno) {

...
}
```



Develop
er



Manage
r

Add
Logging
functionali
ty to all the
methods of
DAO



```

@Repository
public class AccountDAOImpl
    implements
    AccountDAO{
    @Override
    //Logging Code Here
    public void haveAccount(Account
    ...
    }
    @Override
    //Logging Code Here
    public Account getAccount
    ...
    }

```

Add Security Code Here

Add Security Code Here



Develop
er

Add
Security –
Only
Authenticat
ed Users
can access
these
methods



Manage
r



Add Logging and Security Code

Defined with Logging and Security

Controller

Services

DAO Class

Add Logging and Security Code

Understanding Business or Technical Problems

Add Logging & Security code for Services and Controller Classes



Develop



Defined with Logging and Security

Defined with Logging and Security



Defined with Logging and Security

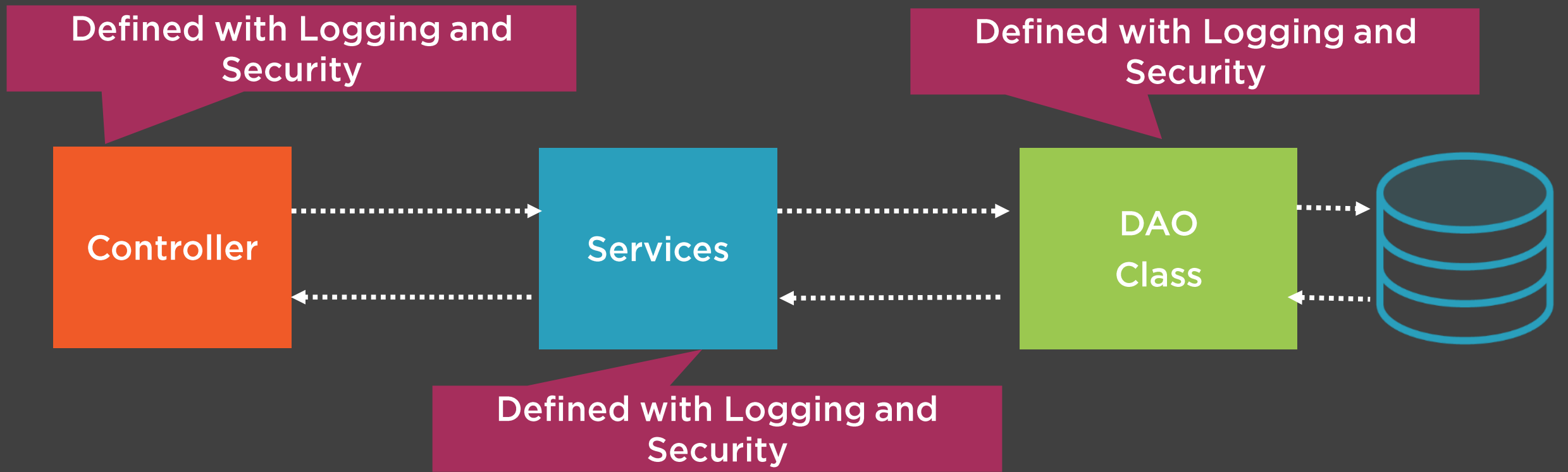
Understanding Business or Technical Problems



Develop

Application consists of 100's DAO, Controllers and Services

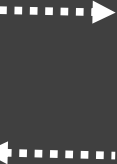




Understanding Business or Technical Problems



Defined with Logging and Security



Defined with Logging and Security

Defined with Logging and Security

Understanding Business or Technical Problems



Develop

Use Aspect Oriented Programming
Popularly Overcome AOP



What is Aspect Oriented Programming?



Aspect Oriented Programming

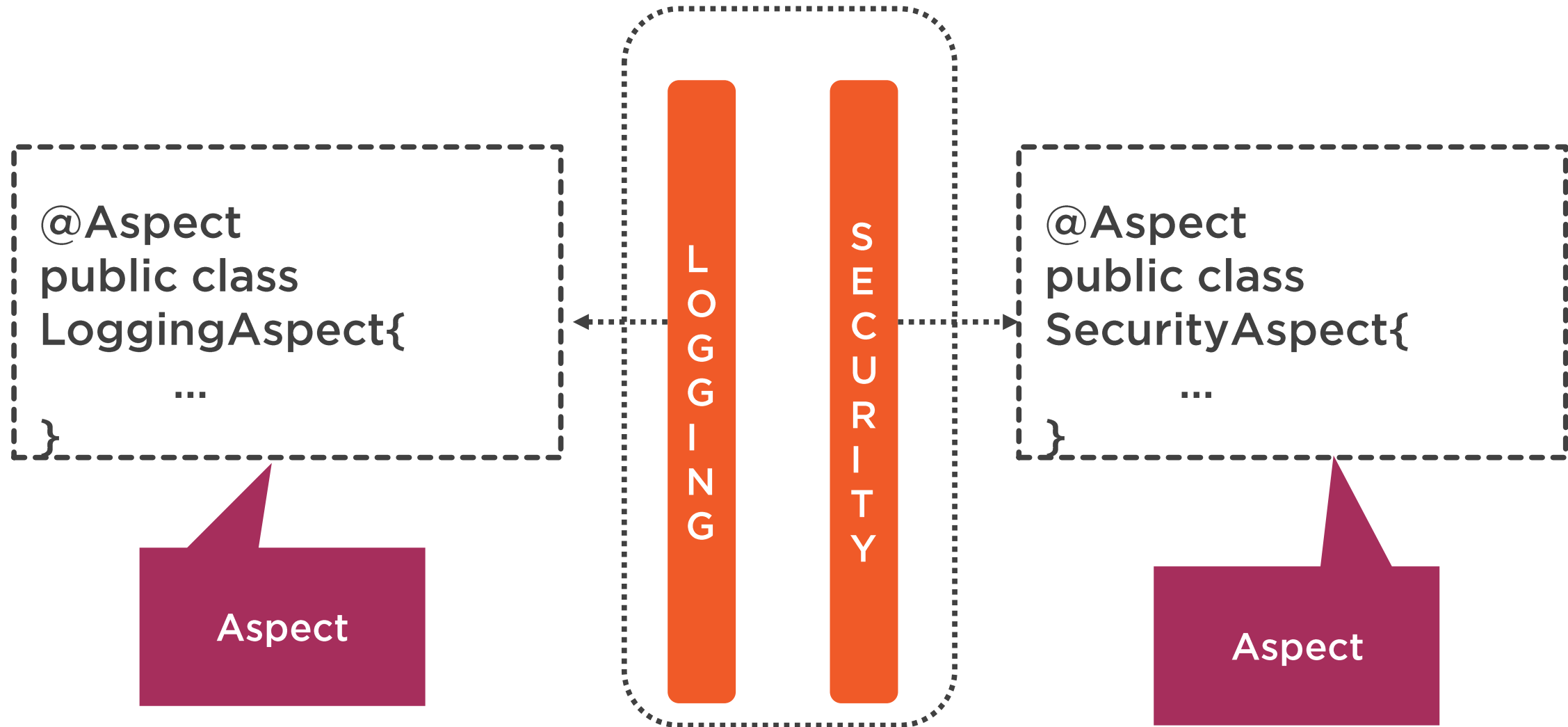
Programming technique based on concept of an Aspect

Aspect encapsulates cross-cutting concerns

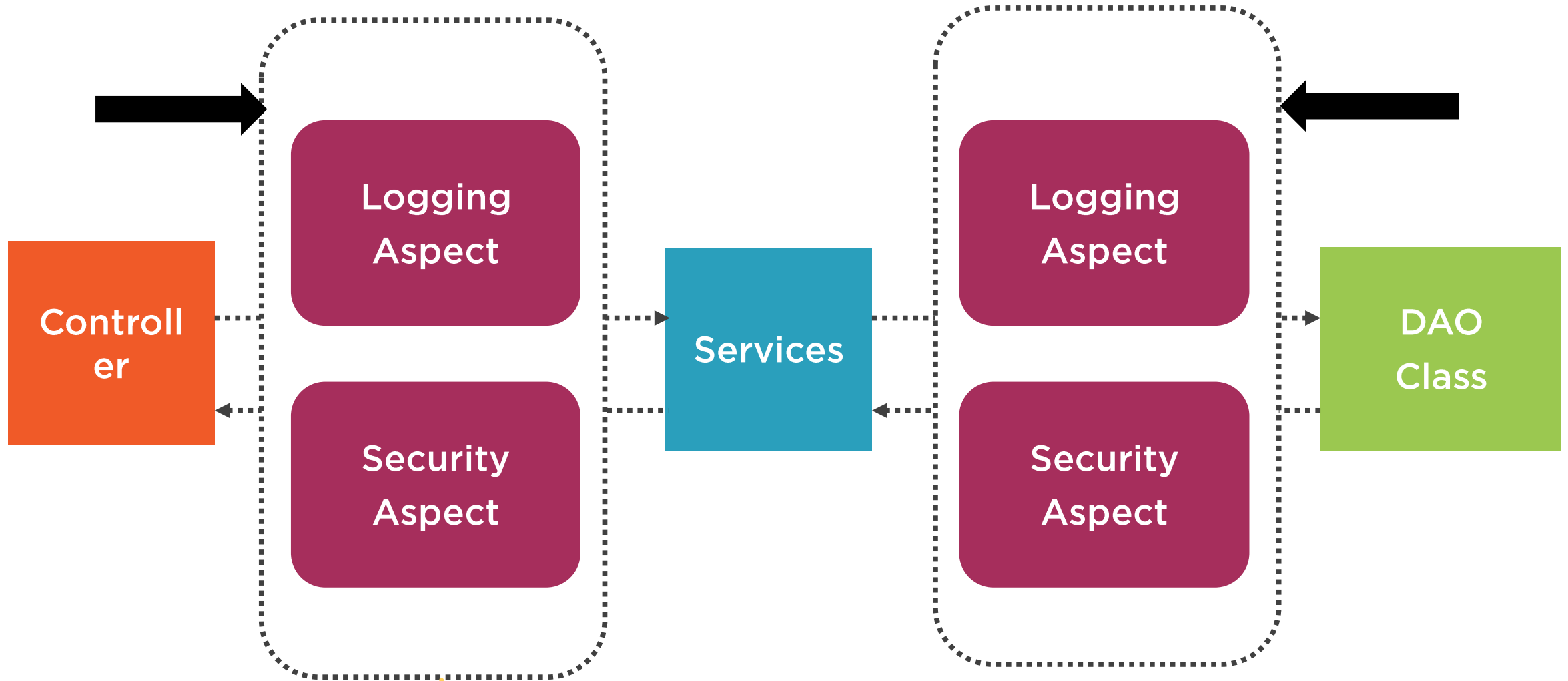
Logic or Functionality



Cross Cutting Concerns



Cross Cutting Concerns



As simple as this application can be, it still has cross-cutting concerns that are common to many projects. A single application can have many different parts for a project.

Understanding AOP Terminology



Understanding AOP Terminology

Aspect

- Is a module of code for cross cutting concerns

Advice



- Uses to specify what action is taken and when it should be applied

Join Point

- Uses to specify when to apply the code during program execution

Pointcut

- Predicate expression to specify where the advice should be applied



Advice Types

Before

- Run the code before the actual method is executed

After Finally

- Run the code after the method finishes its execution

After Returning

- Run the code after the method for a successful execution

After Throwing

- Run the code after the method finishes and if an exception is thrown

Around

- Run the code before and after the method execution



AOP Frameworks

Spring AOP

Aspect J



Comparing Spring AOP and Aspect J



Spring AOP

Advantages

- Very easy to use
- Uses Proxy Pattern
- Easy to migrate from Spring AOP to Aspect J

Disadvantages

- Supports only method level Join points
- Supports only objects created using Spring App Context
- Performance is slow



Aspect J

Advantages

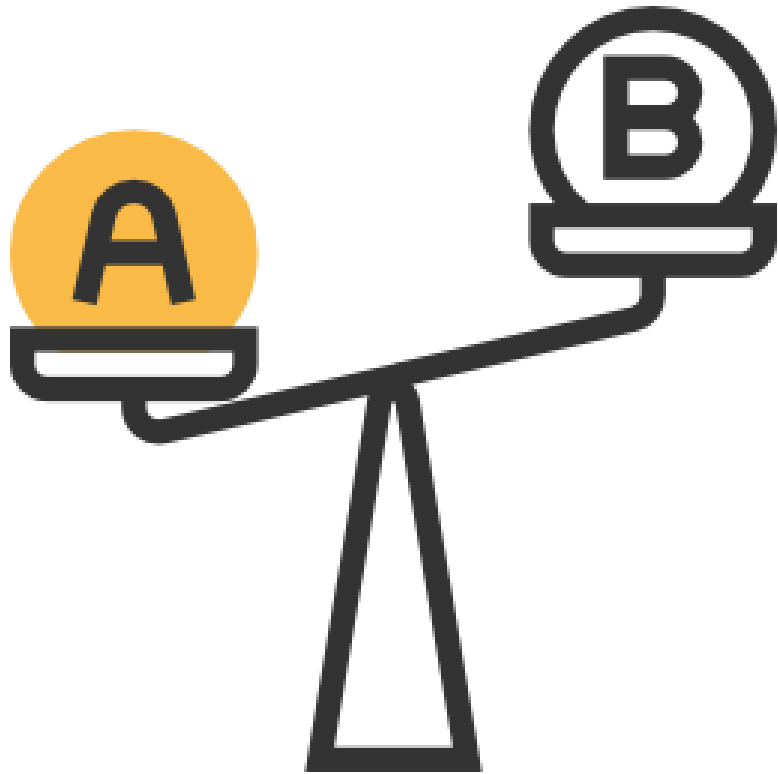
- Supports all Join types
 - Method level, Constructor level and Field level
- Can be used with any POJO
- Performance is faster when compared with Spring AOP
- Provides full implementation of AOP Specification

Disadvantages

- Compile time weaving requires additional compile time step
- Pointcut syntax is complex



Comparing Spring AOP and Aspect J



Spring AOP

- Light implementation of AOP
- Solves only common problems in Enterprise applications

Aspect J

- Very Fast
- Supports Complete AOP functionality required for Enterprise applications
- Complex

Setting up Development Environment for Supporting AOP



Steps to Configure AOP with Spring MVC

①

Download AspectJ Weaver Jar File

②

Copy the AspectJ Weaver Jar file to WEB-INF/lib folder

③

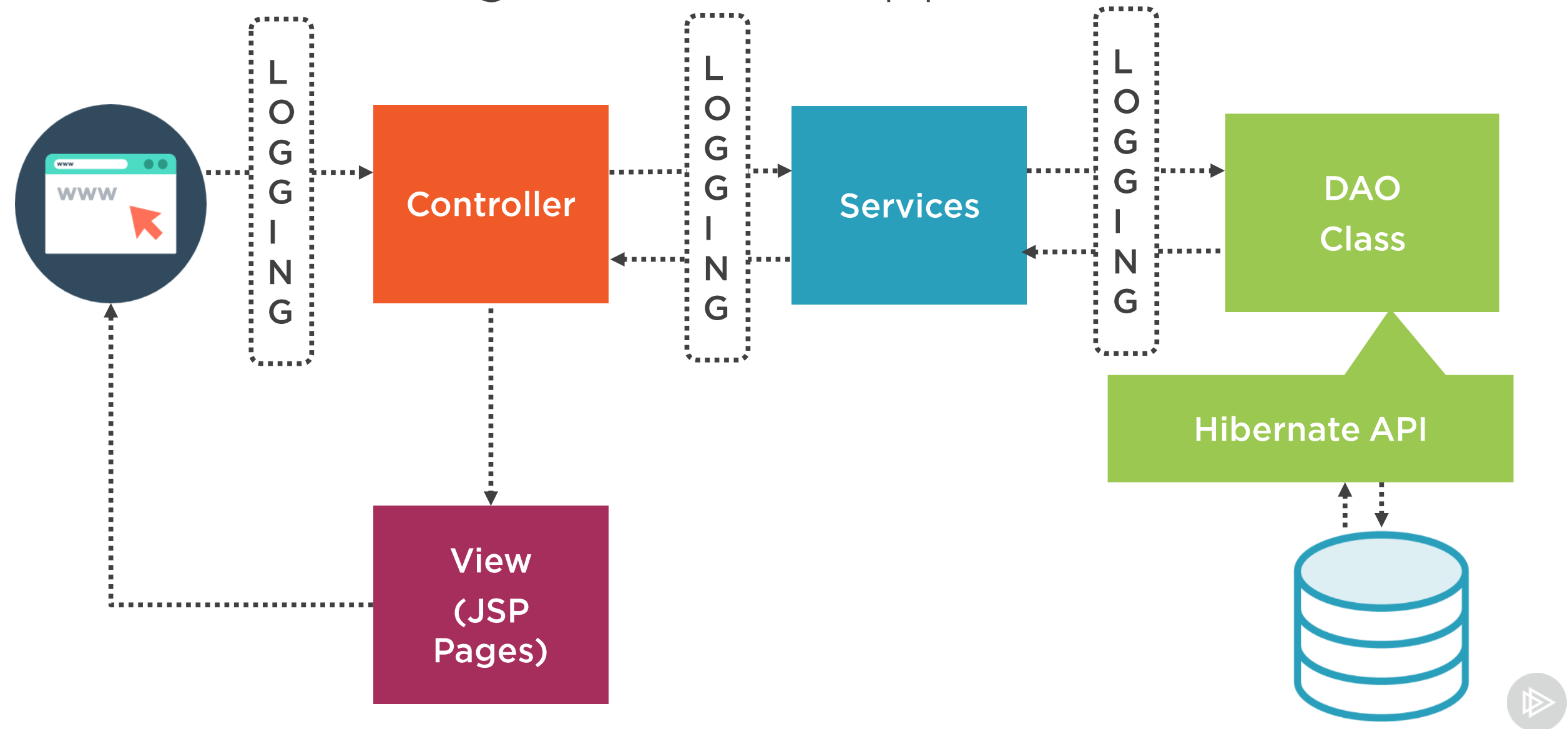
Enable AspectJ Auto Proxy



Demo: Aspect Oriented Programming



Big Picture of Application



Demo: Managing Exceptions Using AOP



Summary



Why Aspect Oriented Programming

Differences between Spring AOP and Aspect J

Set up development environment for supporting AOP

Demo to Log and Manage Exceptions using AOP

