ИСПОЛЬЗОВАНИЕ ASPECTJ В OSGi

AXAMIT

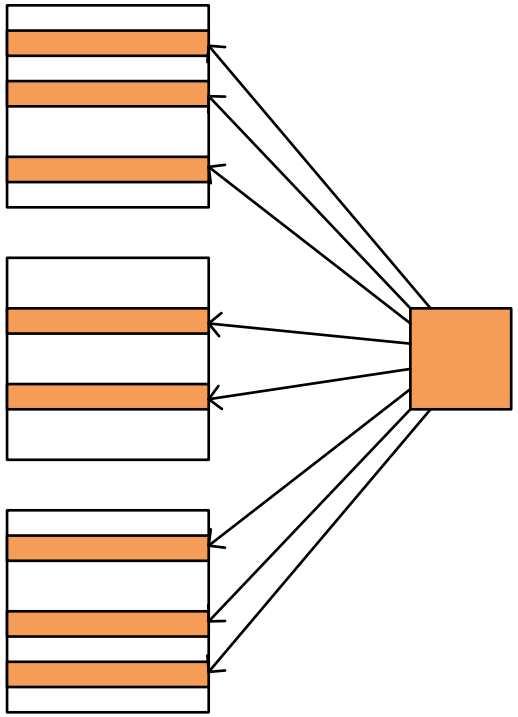
ПЁТР МЕЛЬНИКОВ

AEM Архитектор
Глава R&D в компании Axamit

Email: peter.melnikov@axamit.com

AXAMIT

Что такое АОП и зачем оно нужно?



АОП – это парадигма программирования, позволяющая оформить некоторую общую сквозную функциональность в виде отдельного модуля.

Важное свойство сквозной функциональности это то, что она слабо связана с основной логикой приложения.

AXAMIT

Практическое применение

- Логгирование и аудит
- Кэширование
- Проверка прав доступа
- Транзакции
- Статистика выполнения кода

AXAMIT

Термины АОП

CROSS-CUTTING CONCERN — сквозная функциональность для внедрения.

ASPECT — класс, реализующий сквозную функциональность. Аспект изменяет поведение остального кода в особых точках называемых *Joint Point*, определённых некоторым срезом иначе *Pointcut*.

ADVICE — средство оформления кода, который должен быть вызван для точки соединения. Совет может быть выполнен до, после или вместо точки соединения.

JOINT POINT — это потенциальная соединения точка в потоке выполнения программы, где можно применить *Advice*, чтобы изменить ‘нормальное’ поведение приложения.

POINTCUT — это совокупность всех точек соединения, удовлетворяющих условиям для для внедрения сквозной функциональности, определенной в *Advice*.

WEAVING — или иначе переплетение, это процесс внедрения функциональности в код.

AXAMIT

Почему AspectJ?

AspectJ — это открытое расширение для языка Java, упрощающее реализацию сквозной функциональности для вашего приложения.

AspectJ — де-факто стандарт реализации АОП для языка Java.

Первая версия — в 2001 году

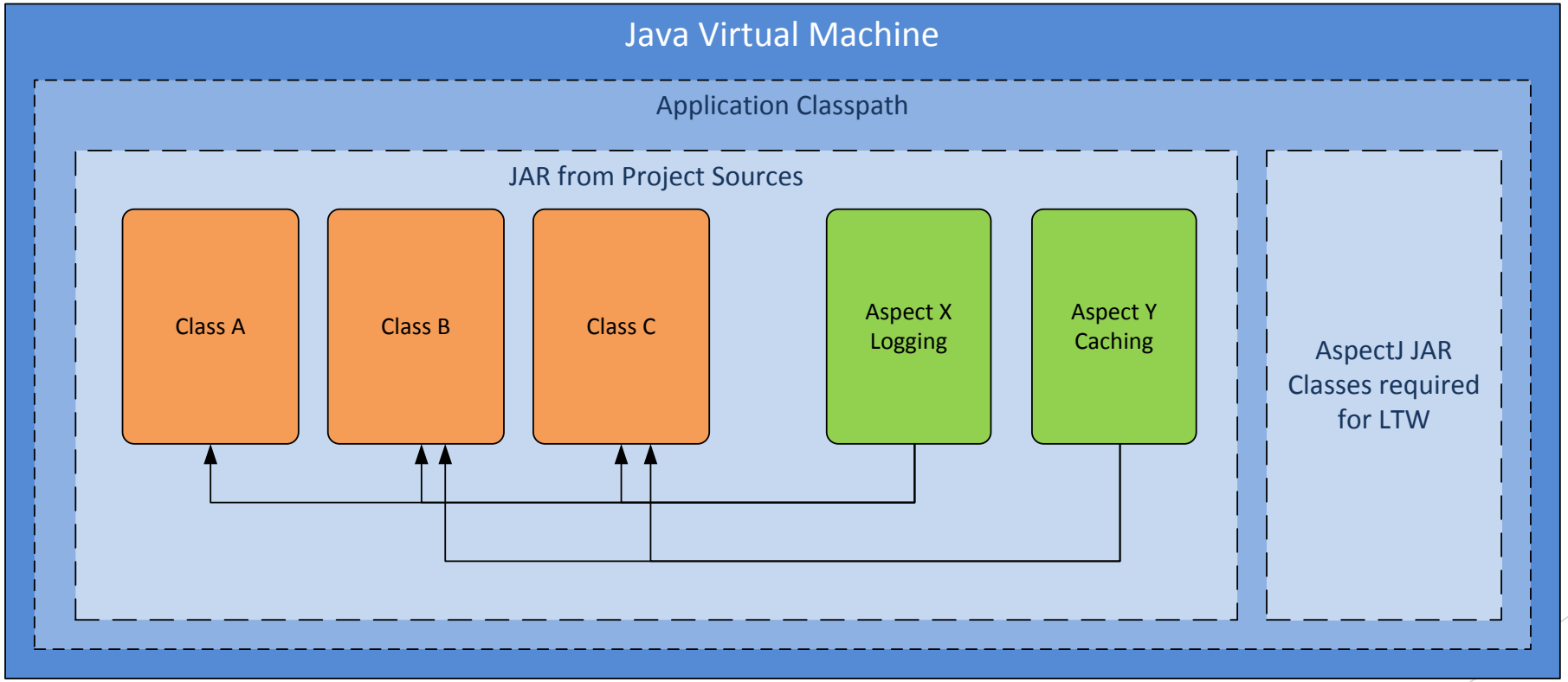
AXAMIT

Пример

```
@Aspect
public class MyAspect {
  ...
  @Pointcut("execution(* *..ClassA.*(..))") // определение Pointcut
  public void allMethodsInClassA() {}

  @Around("allMethodsInClassA()") // определение Around Advice
  public Object aroundMethodsInDemoPackage(
       ProceedingJoinPoint joinPoint) throws Throwable {
     ... // код, который мы хотим выполнить до вызова целевого метода
     String methodName = joinPoint.getSignature().getName()
     LOGGER.info("Before {}", methodName );

     Object result = jointPoint.proceed(); // вызов оригинального метода

     LOGGER.INFO("After {}", methodName );
     ... // код, который мы хотим выполнить после вызова целевого метода
     return result;
  }
}
```

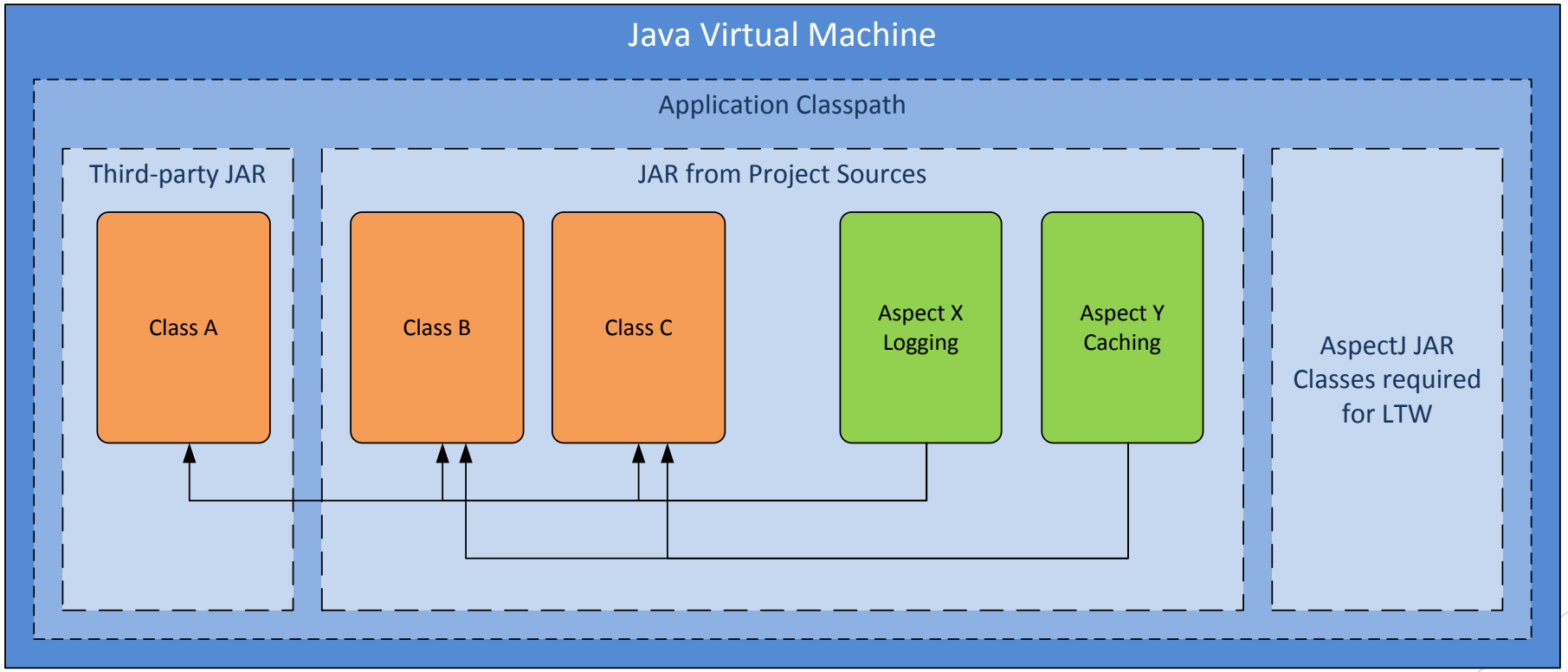
AXAMIT

Сценарии: Базовый



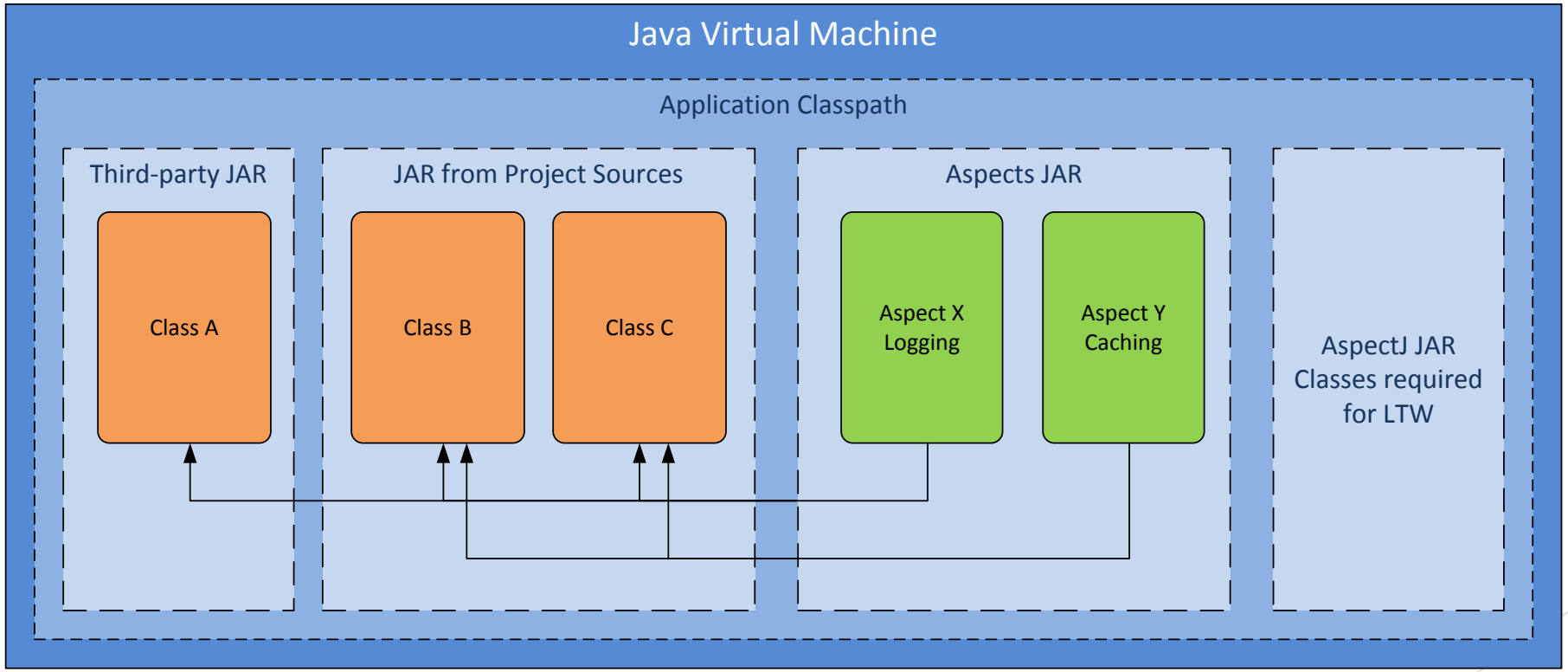
AXAMIT

Сценарии: со сторонним кодом



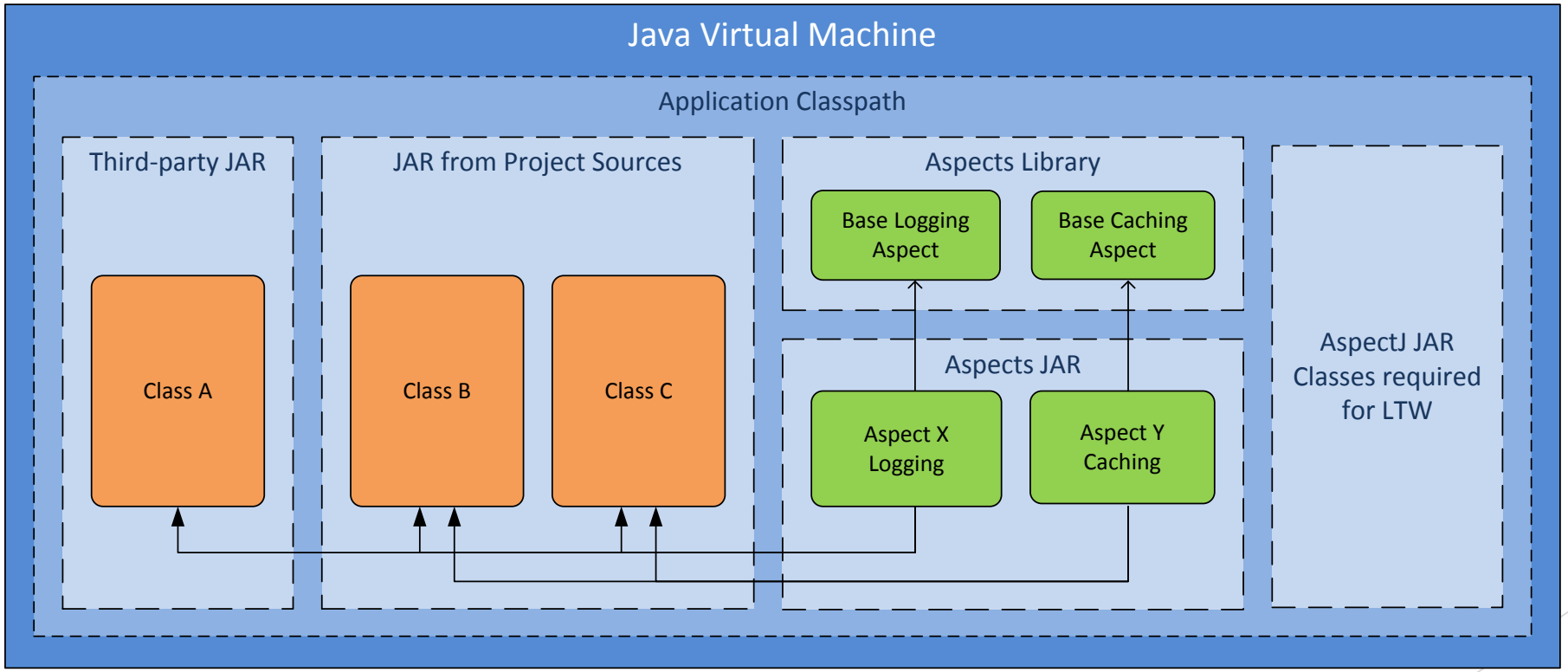
AXAMIT

Сценарии: модуль с аспектами



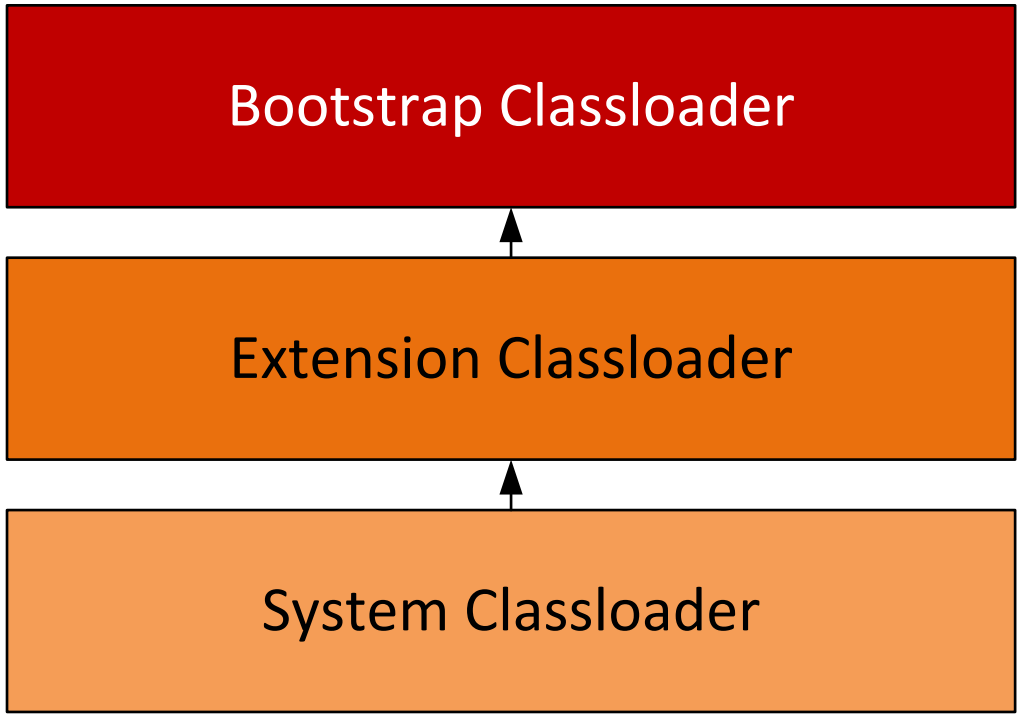
AXAMIT

Сценарии: библиотека аспектов



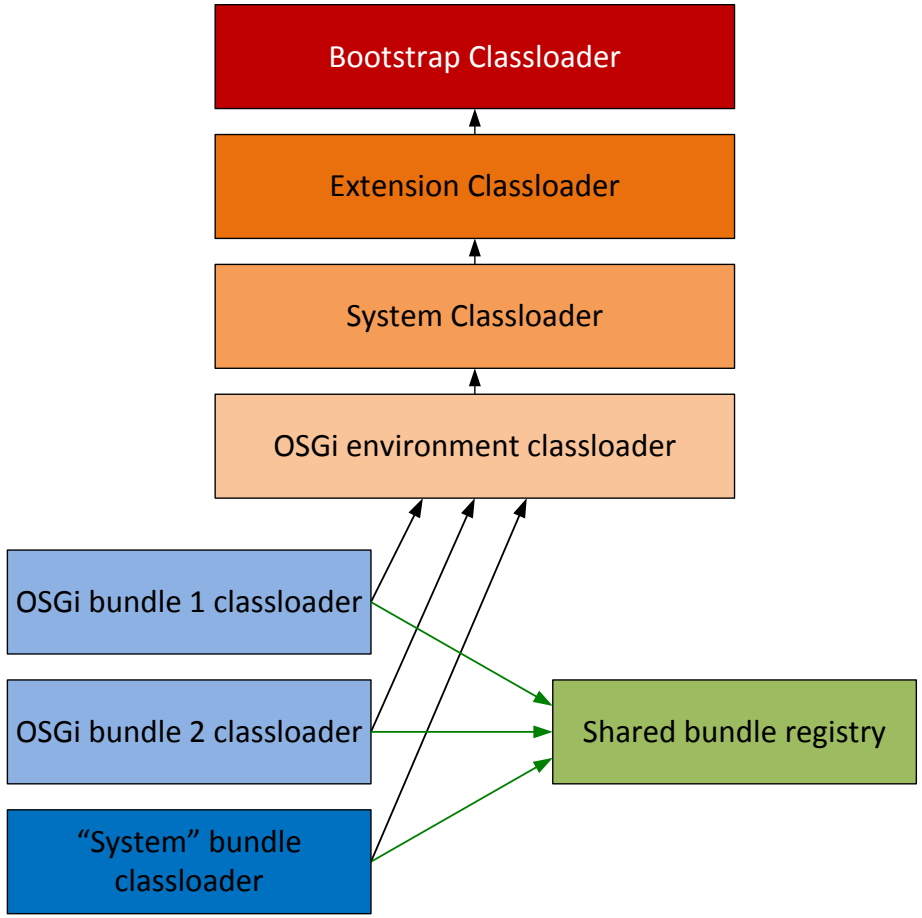
AXAMIT

Стандартная конфигурация классloадеров в Java



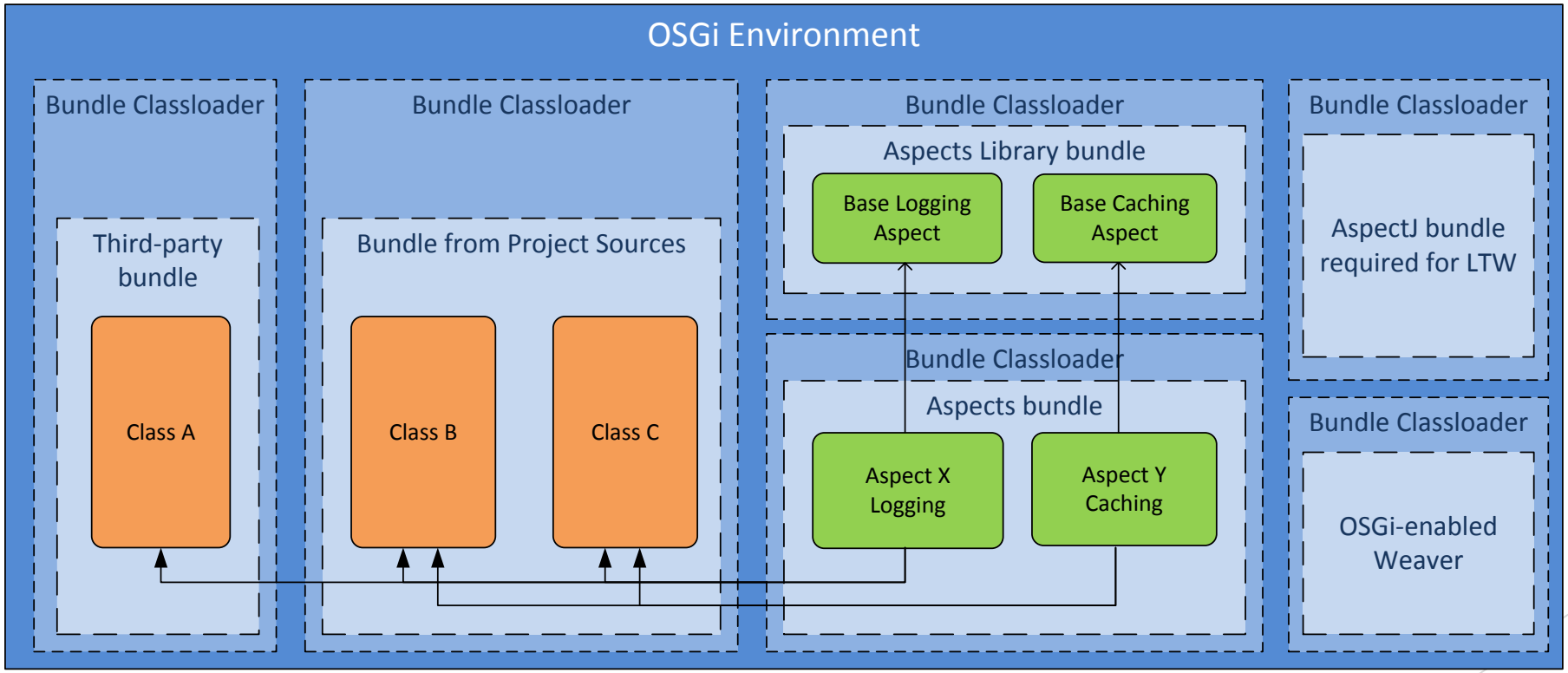
AXAMIT

Иерархия класслоадеров в OSGi



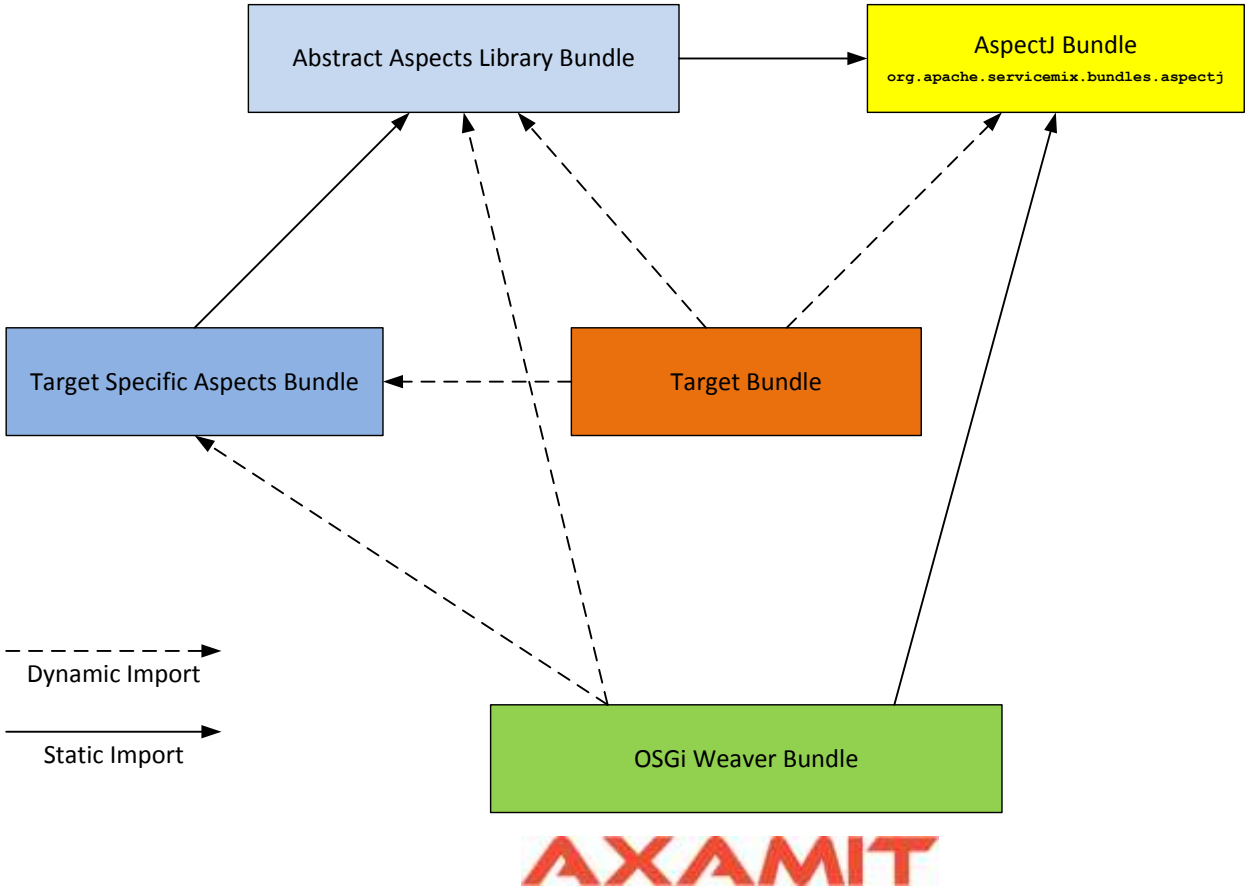
AXAMIT

Проблемы реализации АОП в OSGi



AXAMIT

Диаграмма зависимостей



AXAMIT

Интерцепторы связывания в Apache Felix

```
@Component
@Provides
public class ProxyBindingInterceptor extends DefaultDependencyInterceptor
    implements ServiceBindingInterceptor {

    ...

    @Override
    public <S> S getService(DependencyModel dependency,
                            ServiceReference<S> reference, S service) {
        S proxy = (S) Proxy.newProxyInstance(this.getClass().getClassLoader(),
                new Class[]{ dependency.getSpecification() },
                new DynamicInvocationHandler(service)); // наш обработчик
        deps.put(reference, proxy);
        return proxy;
    }
    ...
}
```

AXAMIT

Интерцепторы связывания в Apache Felix

```
public class TimingDynamicInvocationHandler implements InvocationHandler {
    ...
    public TimingDynamicInvocationHandler(Object target) {
        ...
    }

    @Override
    public Object invoke(Object proxy, Method method, Object[] args)
            throws Throwable {
        long start = System.nanoTime();
        Object result = methods.get(method.getName()).invoke(target, args);
        long elapsed = System.nanoTime() - start;
        LOGGER.info("Executing {} finished in {} ns", method.getName(), elapsed);
        return result;
    }
}
```

AXAMIT

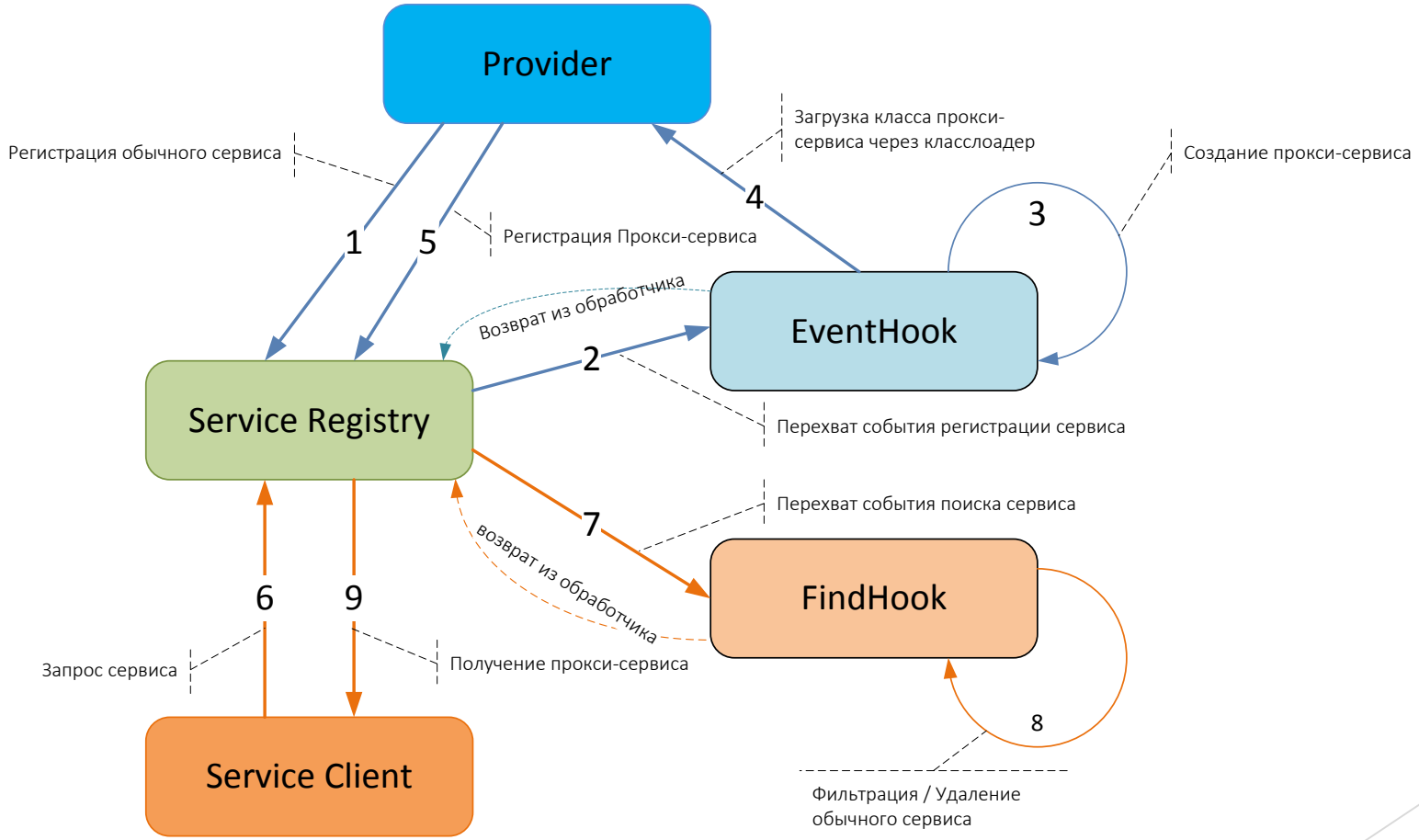
Интерцепторы связывания в Apache Felix

- Не требуется подключать какие-либо дополнительные зависимости.

- Использование интерцепторов завязано на Apache Felix API

AXAMIT

EventHook+FindHook



AXAMIT

EventHook+FindHook

- Используется стандартное API OSGi

- Все еще сложно управлять тем, какие методы в сервисах мы переопределяем

- Мы не можем переопределить функциональность в классах, которые не экспортируются из бандла как сервисы

AXAMIT

Краеугольные камни АОП в OSGi

- Инструментирование классов в OSGi контейнере
- Динамический импорт зависимостей
- Сканирование ресурсов

AXAMIT

Та-дам, встречайте: WeavingHook!

```
package org.osgi.framework.hooks.weaving;
interface WeavingHook {
    void weave(WovenClass wovenClass);
}
```

AXAMIT

Подключение AspectJ

`org.apache.servicemix.bundles.aspectj`

Этот бандл это OSGi обертка библиотек **aspectjweaver** и **aspectjrt**.

aspectjweaver - используется на этапе инструментирования
aspectjrt - нужен во время работы приложения.

AXAMIT

Проблема №1: как найти аспекты?

Просканировать classpath на наличие классов с аннотацией `@Aspect` мы не можем ввиду ограничений OSGi.

AXAMIT

Решение: listResources

```
public interface BundleWiring extends BundleReference, Wiring {

    Collection<String> listResources(String path, String filePattern,
                                     int options);

}
```

path – пакет, преобразованный в путь.
filePattern – '.class'
options – параметры сканирования: сканировать ли рекурсивно и включать ли ресурсы из других бандлов)

AXAMIT

Проблема №2: полное сканирование медленное и неэффективное

Нет большого смысла перебирать все загруженные бандлы

AXAMIT

Решение: указываем список бандлов

- создаем конфигурацию со списком бандлов с аспектами
- если бандл с аспектами еще не стартовал, принудительно его стартуем, иначе не сможем просканировать
- кэшируем результат поиска аспектов: **class URL + bundle id**

AXAMIT

Проблема №3: AspectJ weaver не видит классов

Должны быть доступны:

- Классloадер с целевым классом, который будет инструментироваться
- Классloадер с аспектами и `aop.xml` конфигурацией
- Классloадер с библиотекой абстрактных аспектов

AXAMIT

Решение

`CompositeClassLoader` - объединяет все классслоадеры в один и умеет делегировать загрузку классов этим классслоадерам

AXAMIT

Проблема №4: AspectJ не находит aop.xml

- `META-INF/aop.xml` - стандартная локация для конфига
- Не получится экспортировать из бандла, так как путь не является пакетом

`aop.xml:`

```
<aspectj>
    <weaver options="-verbose -showWeaveInfo">
    </weaver>
    <aspects>
        <aspect name="com.axamit.aop.logging.CustomLoggingAspect" />
    </aspects>
</aspectj>
```

AXAMIT

Решение: перемещаем aop.xml

- Перемещаем `aop.xml` в экспортируемый пакет, кладем в `/resources/`**`org/aspectj -> org.aspectj`**
- Переопределяем метод `getDefinitions` в `IWeavingContext`, там где происходит парсинг

AXAMIT

Проблема №5: ClassNotFoundException

- Не находятся классы из бандла AspectJ
- Не находятся классы аспектов при вызове методов инструментированного класса

AXAMIT

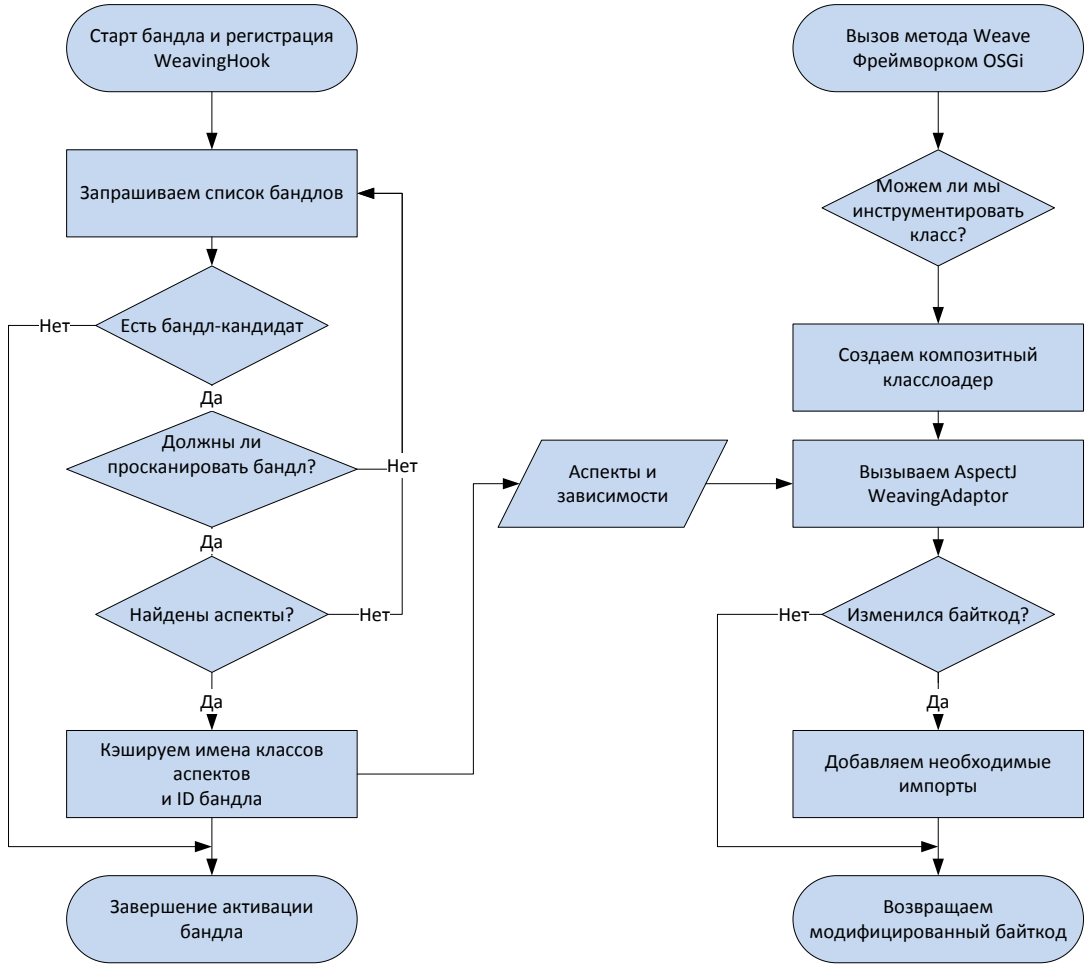
Решение: делаем динамический импорт

org.osgi.framework.hooks.weaving.WeavingHook:

```
public void weave(WovenClass wovenClass) {
    ...
    byte[] originalClassBytes = wovenClass.getBytes(); // original bytecode
    byte[] wovenClassBytes = weavingAdaptor.weaveClass(
wovenClass.getClassName(), originalClassBytes); // enhanced class' bytecode
    wovenClass.setBytes(wovenClassBytes); // update class bytecode
    List<String> imports = wovenClass.getDynamicImports();
    imports.add("aj.org.objectweb.asm"); // add imports
    ...
}
```

AXAMIT

Упрощенная блок-схема



AXAMIT

TODOs

- Реализовать корректную обработку при деактивации и рестарте бандлов
- Научиться восстановать исходный байткод при деактивации аспектов
- Вынести все настройки в конфигурацию
- Прикрутить Java Cache API

AXAMIT

DEMO

AXAMIT

Полезные ссылки

- https://dzone.com/articles/osgi-service-hook-log-all
- https://www.slideshare.net/mfrancis/bytecode-weaving
- http://www.martinlippert.org/events/WJAX2008-AspectWeavingOSGi.pdf
- https://www.ibm.com/developerworks/ru/library/j-aopwork15/index.html

AXAMIT