



Experiences using Equinox Aspects in a real-world project

Heiko Seeberger (metafinanz GmbH)
Martin Lippert (akquinet it-agile GmbH)



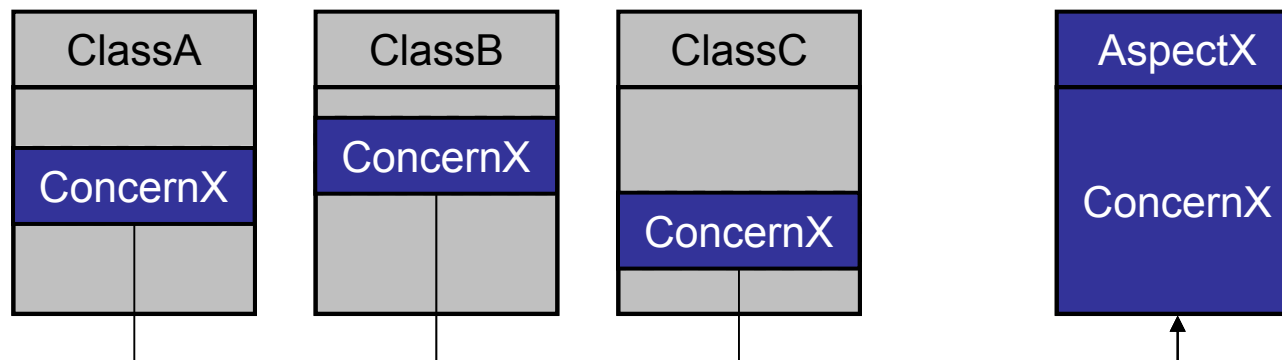
Overview

- Quick overview of AOP
- AOP in the OSGi world?
 - ◆ What does it mean?
- Equinox Aspects: Aspect Weaving for OSGi
 - ◆ Introduction
 - ◆ Demo
 - ◆ Allianz Business System
- What's next for Equinox Aspects?
- Conclusions



Aspect-oriented programming

- Modularity improved a lot by OO concepts
 - ◆ Classes, interfaces
 - ◆ Information hiding, polymorphism, inheritance
- AOP adds additional concepts
 - ◆ To modularize so called “cross-cutting concerns”





AOP today

- Meanwhile AOP is an established concept
 - ◆ Useful for many situations
 - ◆ Mostly technology-centric usage scenarios
- Established languages and frameworks available
 - ◆ AspectJ: powerful language extension to Java
 - ◆ Spring-AOP: simple to use AOP for enterprise apps
- Used in production:
 - ◆ Spring itself makes heavy use of AOP concepts
 - ◆ App-servers are using AOP inside
 - ◆ Direct AOP selectively used in enterprise apps

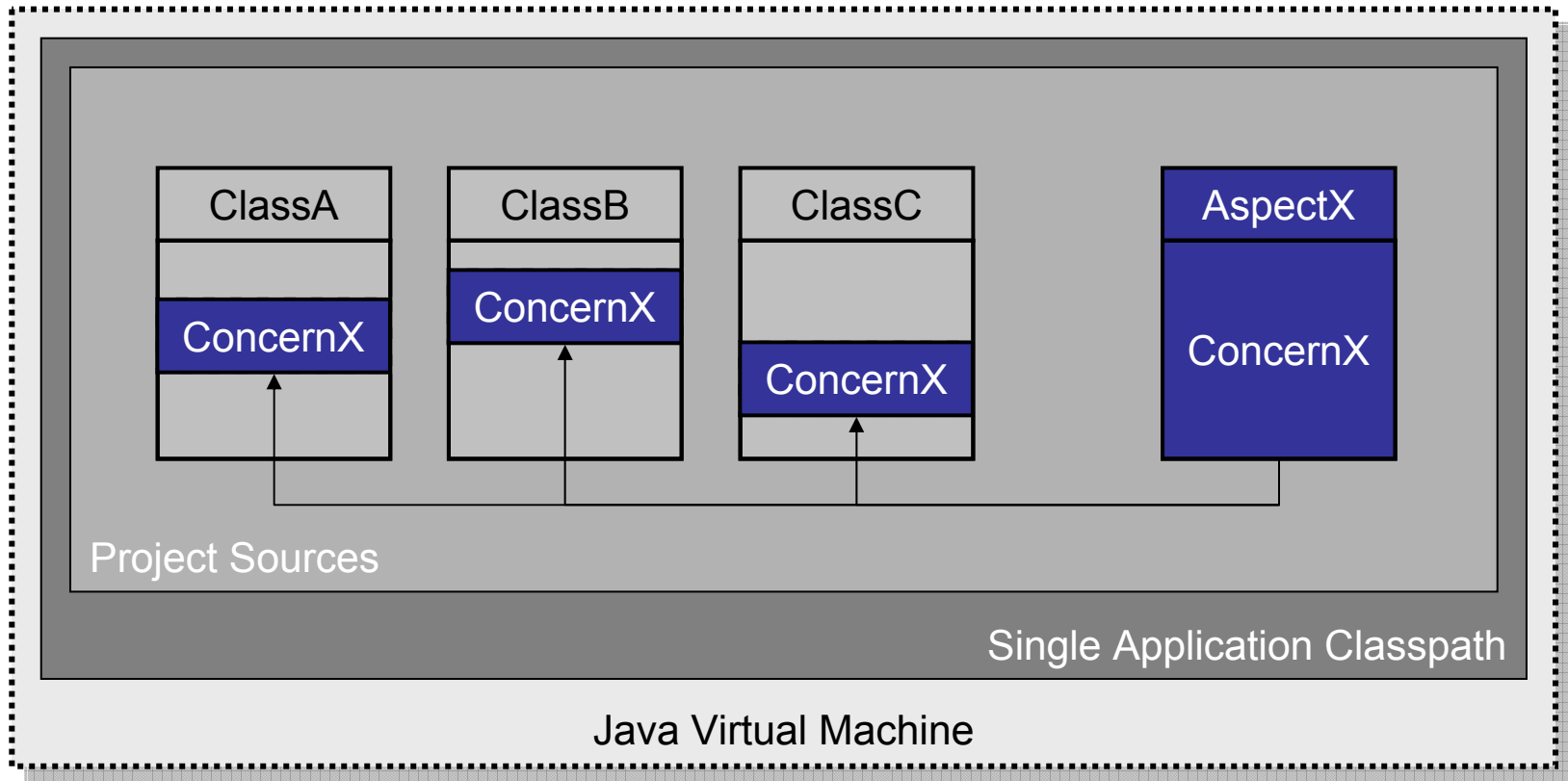


AspectJ = AOP for Java

- AspectJ is a powerful language extension for Java
 - ◆ Hosted as an Eclipse project
 - ◆ Still very active (latest release Dezember 2007, upcoming release 1.6.0 close)
- AJDT:
 - ◆ Great tooling for the Eclipse IDE
 - ◆ Comes close to the JDT feeling
- Spring-IDE:
 - ◆ Integrates AJDT with Spring-AOP
 - ◆ AJDT feeling for Spring apps

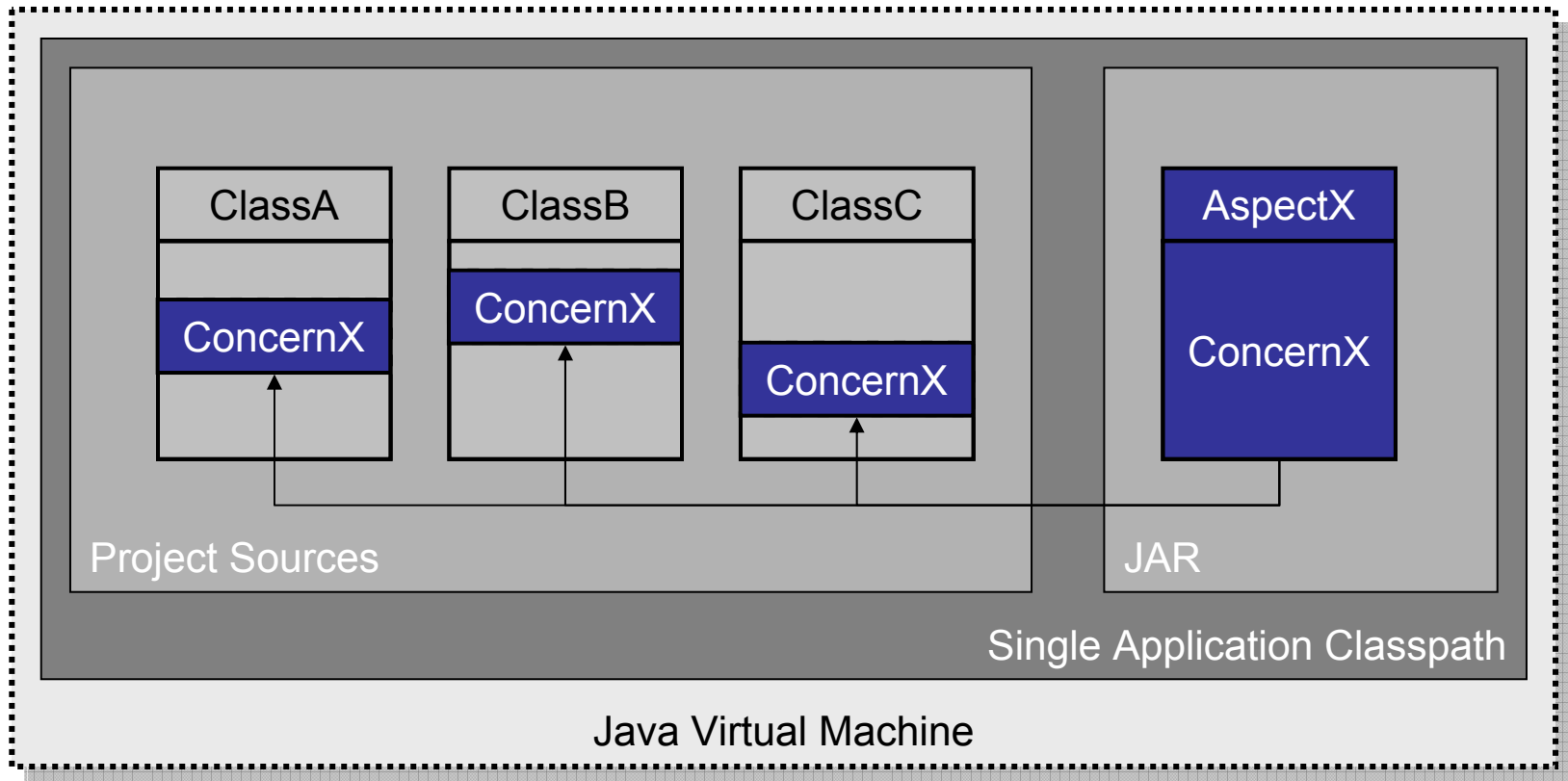


The Standard Use Case



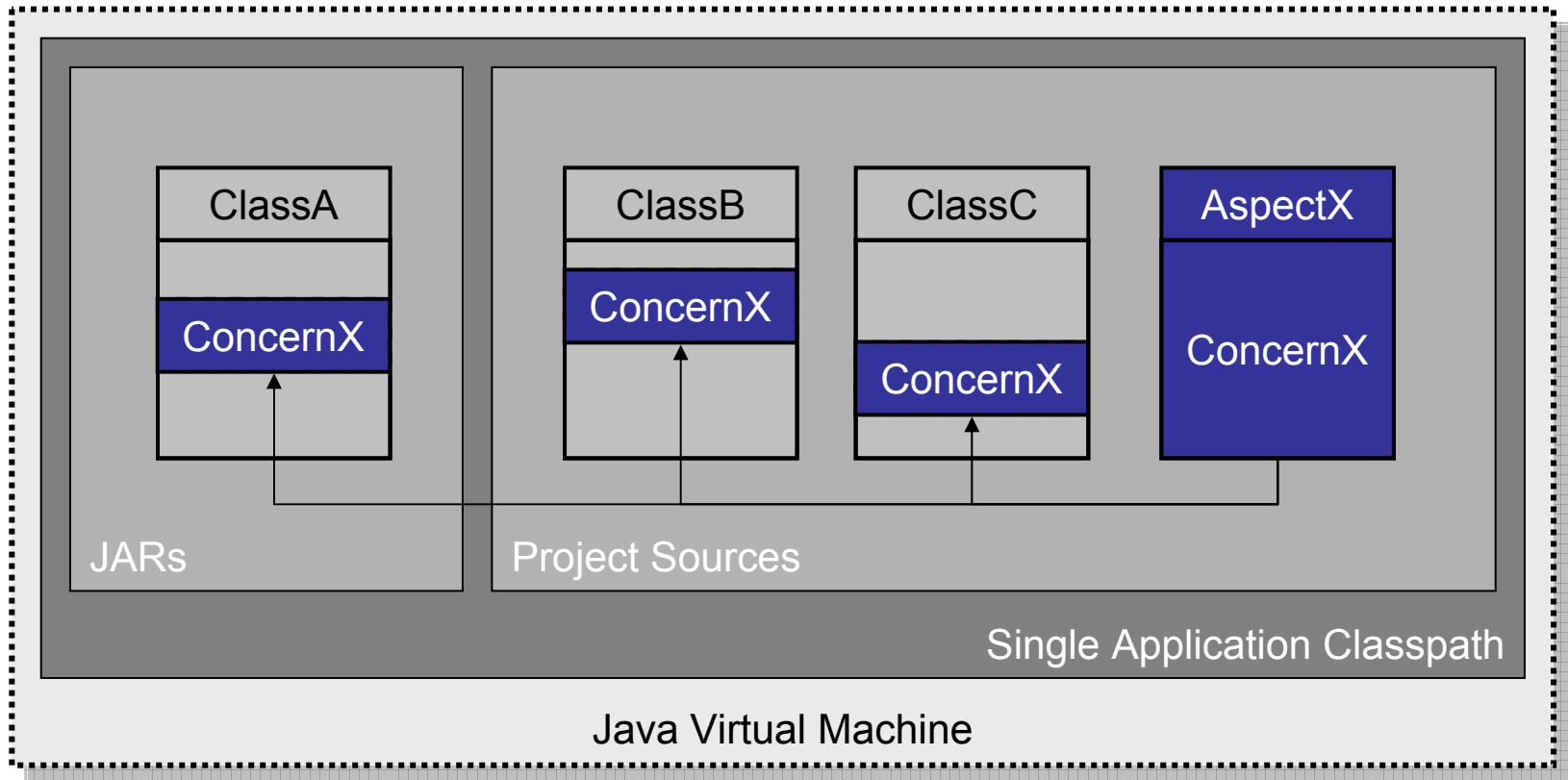


Library Aspects





Aspects for Existing Code





Java™ + OSGi™

- **OSGi™:**
 - ◆ **“A dynamic module system for Java™”**
- Modularity
- Dynamic
- Service-Oriented

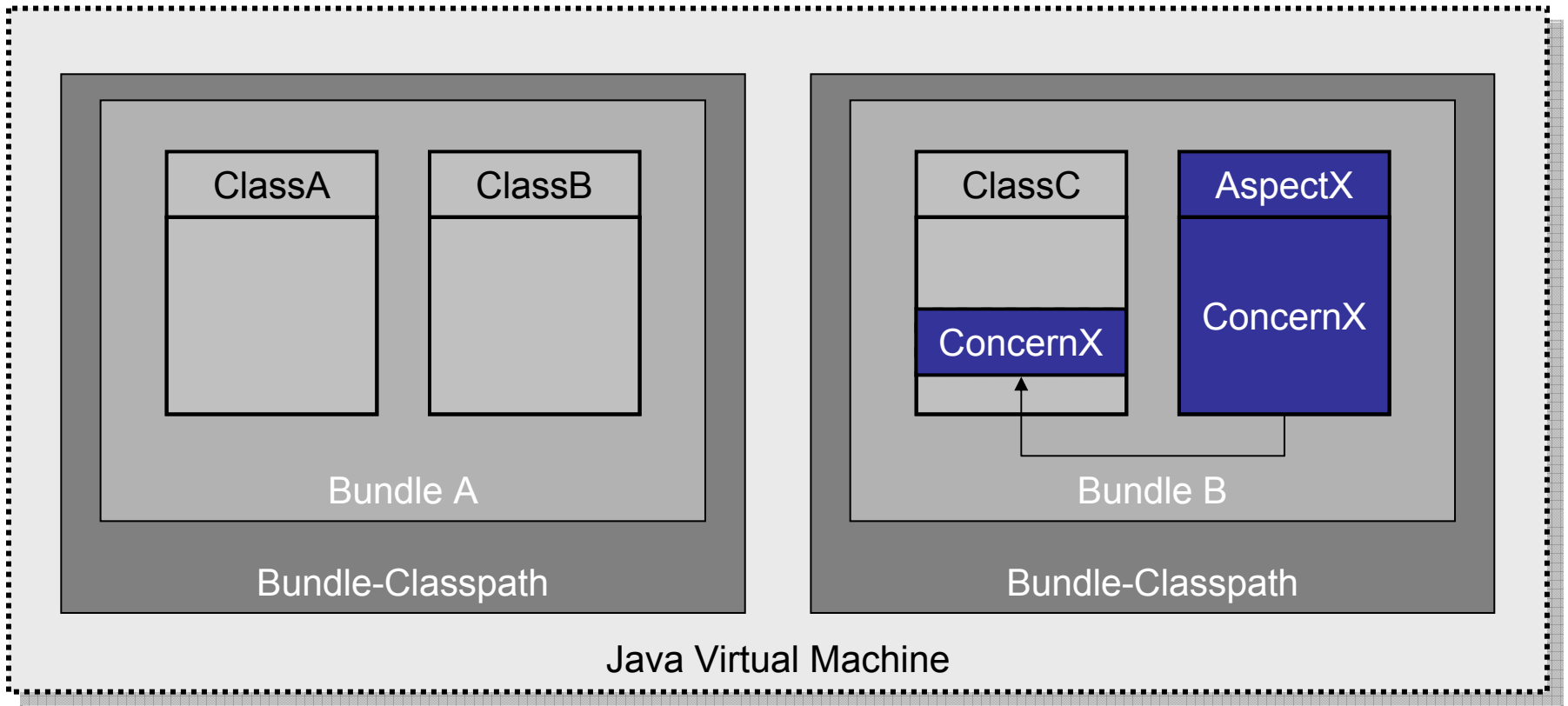


What does it mean for us?

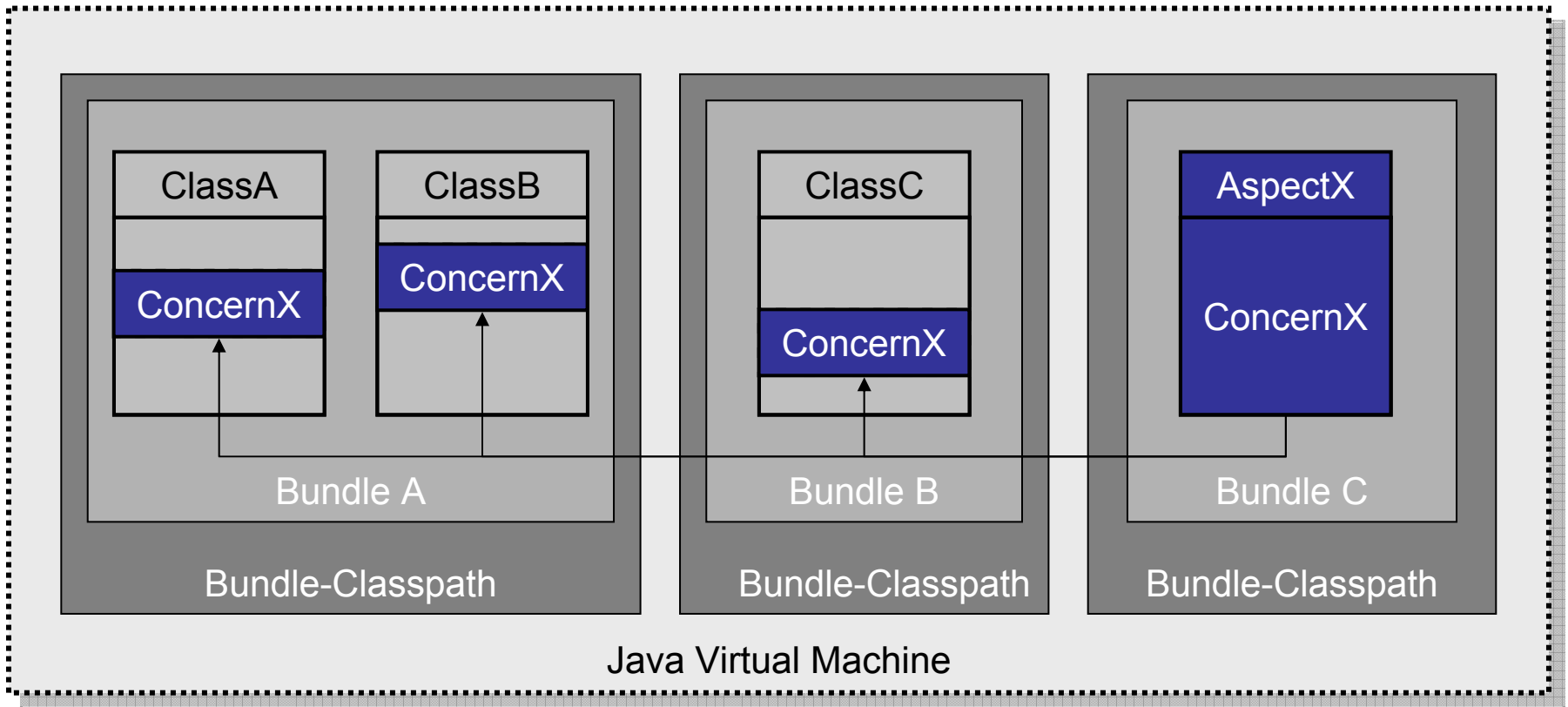
- We would like to **modularize**
 - ◆ ... classes and interfaces into bundles
 - ◆ ... **and** aspects into bundles
- The obvious next step:
 - ◆ **modularize cross-cutting concerns into bundles**
- Takes modularity to the next level



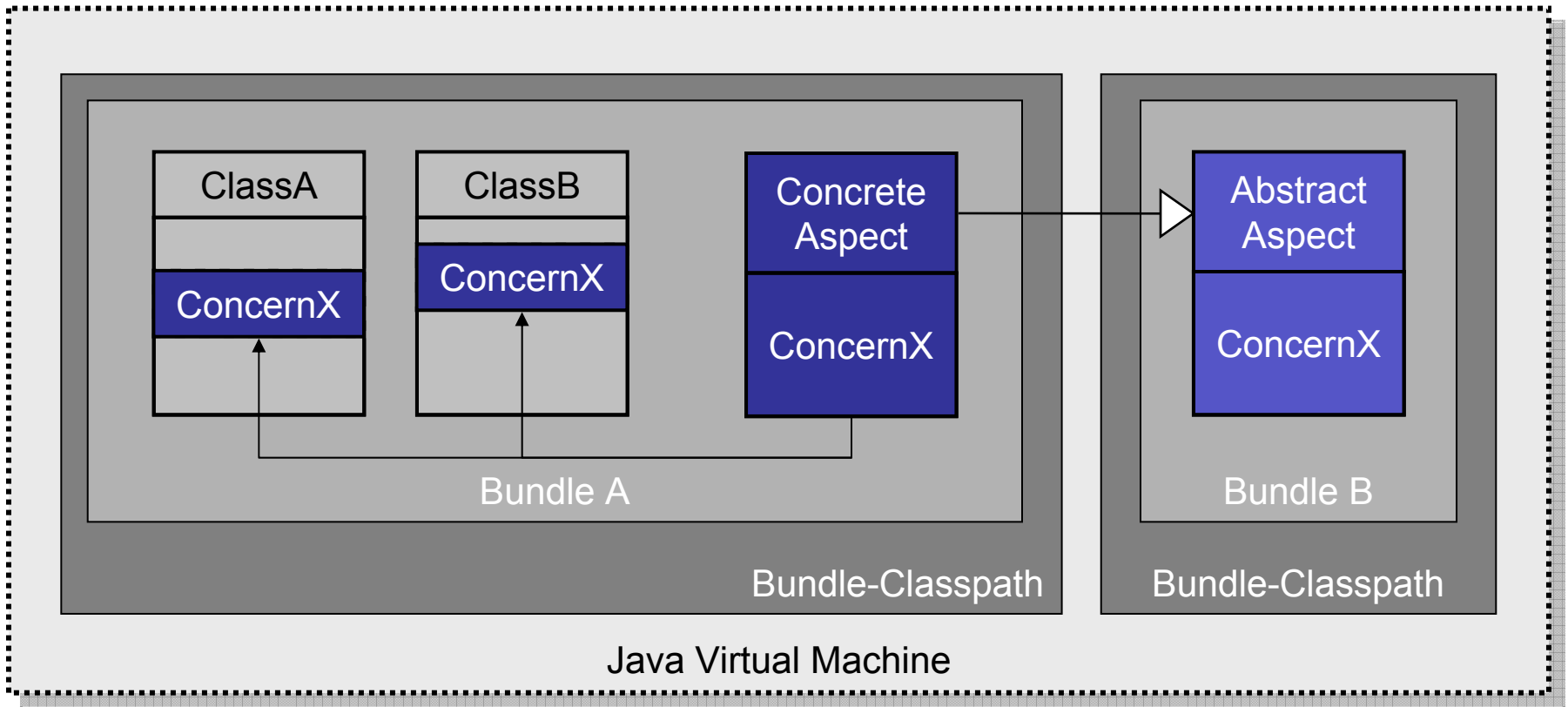
Intra-Bundle Aspects



Co-Op Bundle Aspects



Abstract Aspect Bundles





Dynamics for Aspect Bundles

- OSGi allows dynamic bundle
 - ◆ ... installs
 - ◆ ... uninstalls
 - ◆ ... updates
- Same should be possible for aspect bundles
 - ◆ ... dynamic installs, uninstalls and updates of aspect bundles
 - ◆ ... dynamic installs, uninstalls and updates of bundles that are affected by aspect bundles



Equinox Aspects

- Equinox Incubator Project
 - ◆ <http://www.eclipse.org/equinox/incubator/aspects>
- Enables AspectJ/AOP usage for OSGi
 - ◆ Supports all presented use-cases
 - ◆ Ready-to-use
- Setting
 - ◆ Works with Eclipse 3.3 and 3.4
 - ◆ Works with AJDT 1.5.1, 1.5.2, 1.6.0



Features

- Load-time aspect weaving
 - ◆ No recompilation of existing bundles necessary
 - ◆ Supports full aop.xml load-time weaving config of AspectJ
- Caching for woven classes
 - ◆ Load-time weaving happens only once
 - ◆ Second time startup is same as without aspects
 - ◆ Available for standard JREs and IBM J9 shared classes
 - ◆ Supports configuration switching
- Dynamic installation of aspect bundles
 - ◆ Silent update of bundles to be woven again
 - ◆ Those bundles must behave nicely within dynamic situations



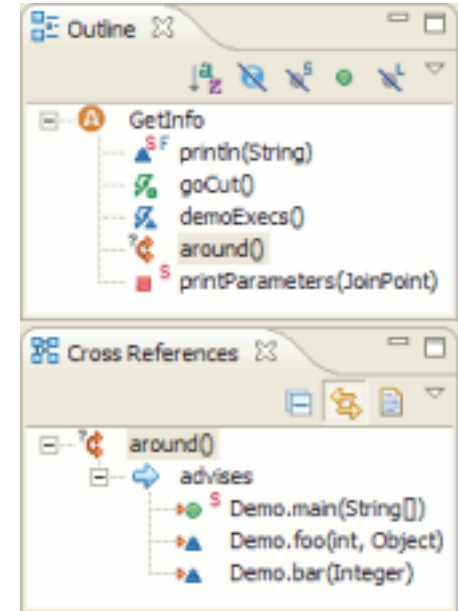
How to use Equinox Aspects?





Install AJDT

- AJDT contains:
 - ◆ AspectJ runtime libraries.
 - ◆ AspectJ integration into SDK (tooling).
- Supported versions (Eclipse 3.3):
 - ◆ 1.5.1 → AspectJ 1.5.4
 - ◆ 1.5.2 → AspectJ 1.6
- www.eclipse.org/ajdt





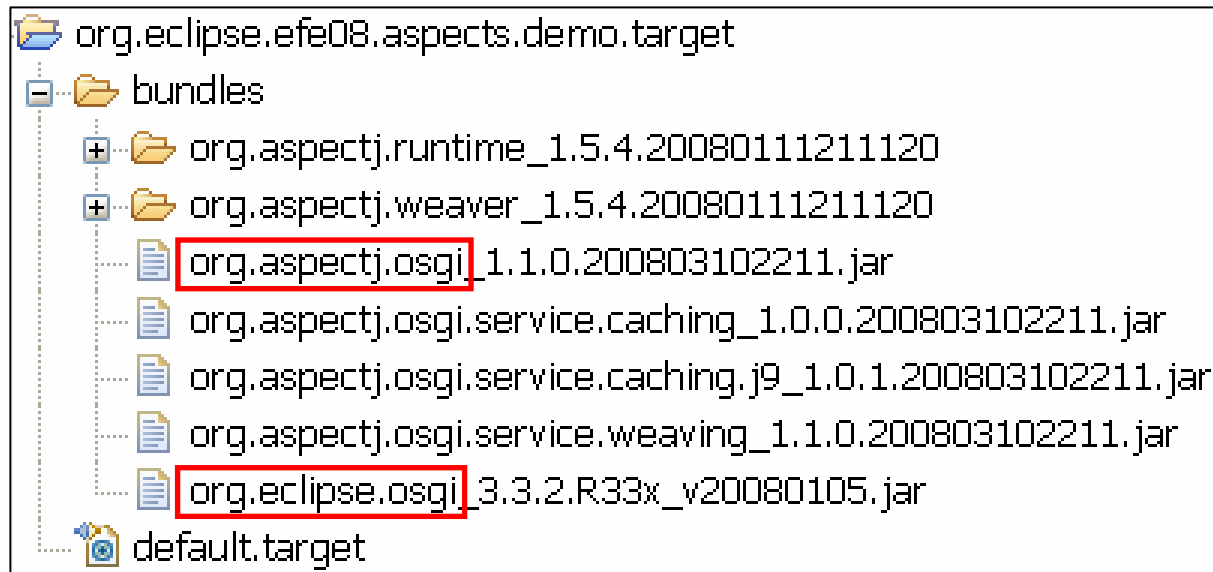
Install Equinox Aspects



- Download Equinox Aspects 1.1.0 from www.eclipse.org/equinox/incubator/aspects
- **Install** into Eclipse installation directory, do not use an extension location!
- Alternatively define a **target** which co-locates `org.aspectj.osgi` and `org.eclipse.osgi`.



Co-locate org.*.osgi





How to write an aspect bundle?

- “Hello world!” on Equinox Console:

```
osgi> hello  
Hello world!
```

- Aspect shaking the output:

```
osgi> hello  
d ovr!Hle
```





How to write an aspect bundle?

- Create a plug-in project.
- Add AspectJ nature.
- Write your aspects.
- Export packages containing aspects.



Demo ...



How to write an aspect bundle?

- Define your aspects in aop.xml.
- Export packages containing aop.xml files.
- Re-export dependency on org.aspectj.runtime bundle.



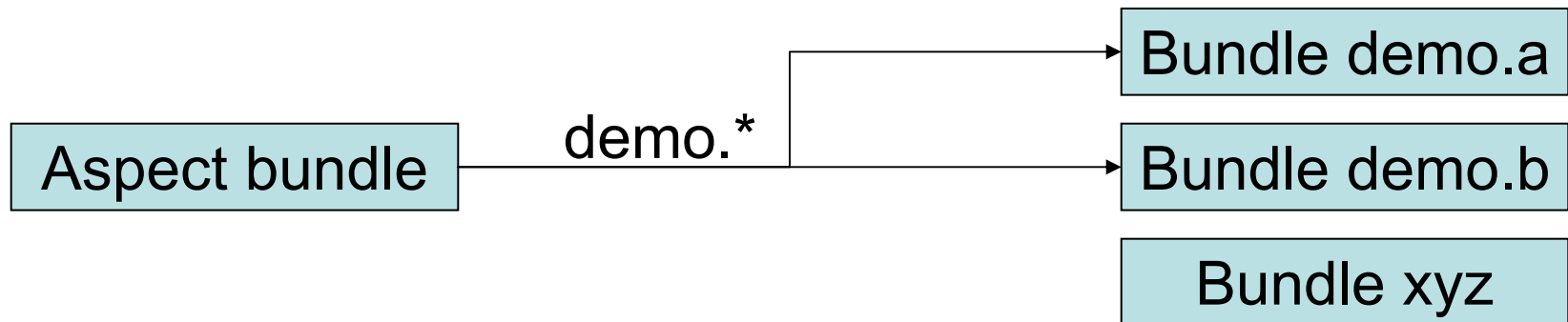
Demo ...



How to write an aspect bundle?

- The aspect bundle must define the “target”:
 - ◆ Which bundles are to be woven?
- Use Eclipse-SupplementBundle header.

```
Eclipse-SupplementBundle: demo.*
```



Demo ...



How to configure Equinox Aspects?



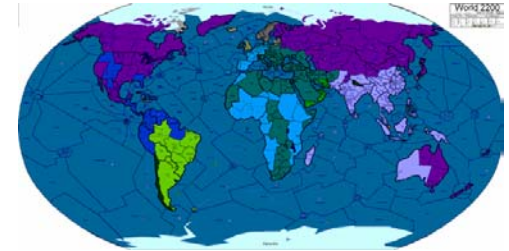
- Tell Equinox to use the Equinox Aspects framework extension:

```
-Dosgi.framework.extensions=org.aspectj.osgi
```

- Declare where to find aop.xml-files:

```
-Dorg.aspectj.weaver.loadtime.configuration=  
org/aspectj/aop.xml
```

Demo ...

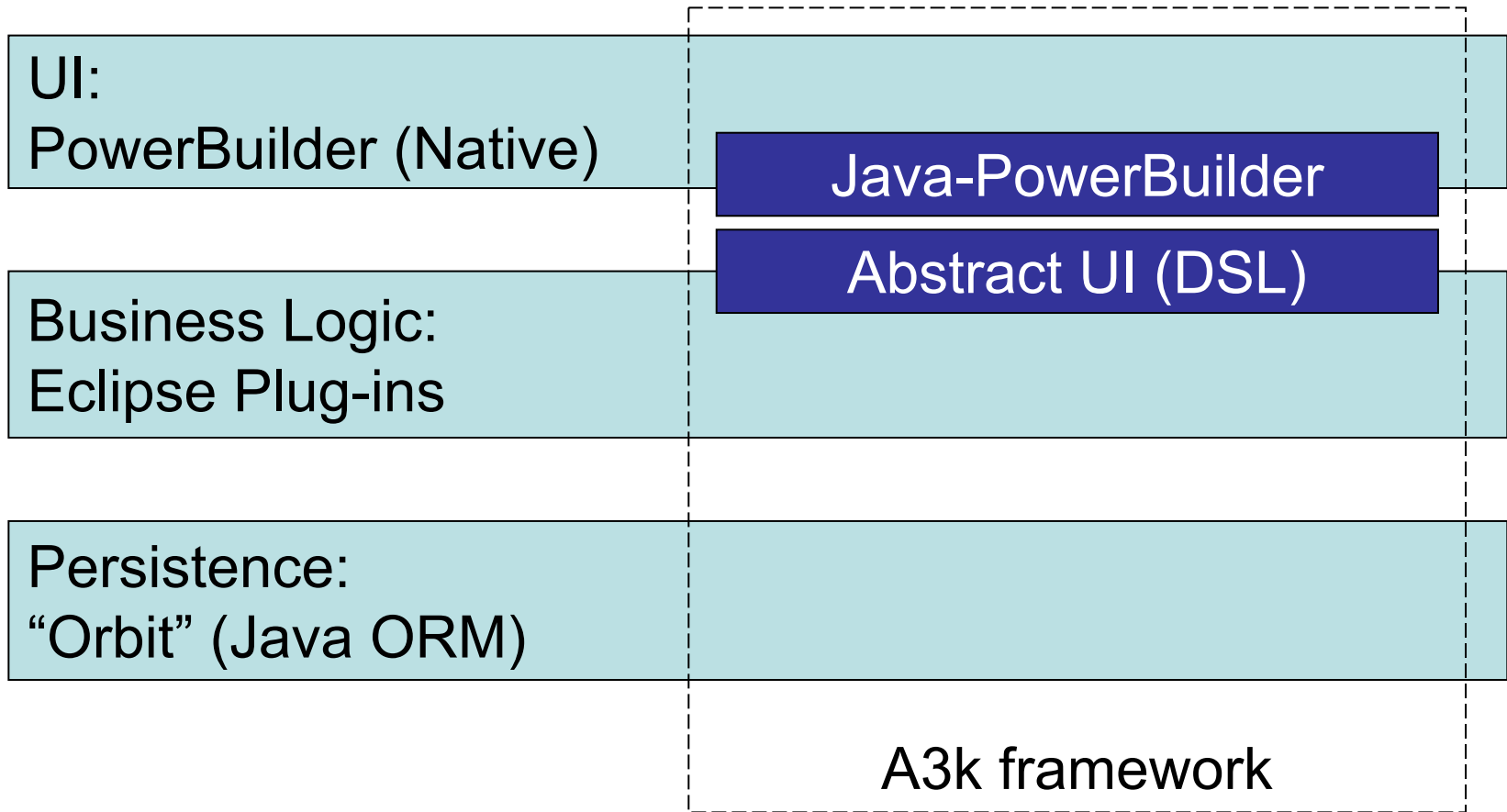


Is there a real-world project?

- Allianz Business System (ABS)
 - ◆ Mission critical system at the core of the insurance business.
 - ◆ Addresses all major concerns in the various classes of insurance.
 - ◆ Online, offline and integration scenarios.
- ABS Client
 - ◆ Rich Client application for
 - call center and
 - clerks.



ABS Client architecture





Top requirements for mission critical systems





How to ensure performance?



Performance logging!



Performance logging

- **Define** service level agreements for critical processes.
 - ◆ E.g.: “Customer-search must show result within two seconds.”
- **Measure** the execution of these processes.
- **Analyze** and **report** the measured data.

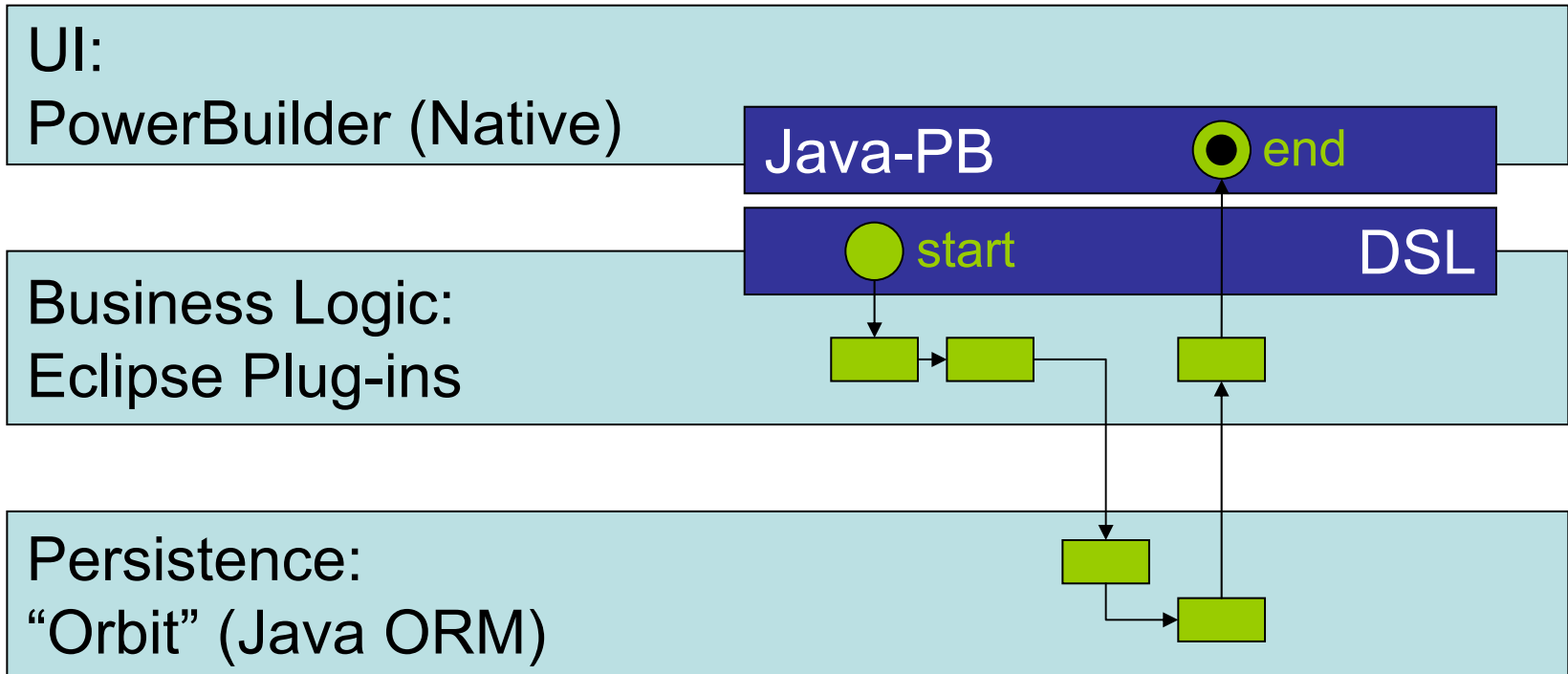


Performance logging

- Define service level agreements for critical processes.
 - ◆ E.g.: “Customer-search must show result within two seconds.”
- **Measure** the execution of these processes:
 - ◆ Log start time.
 - ◆ Log end time.
- Analyze and report the measured data.



Challenge: Performance Logging





Challenge: Performance Logging



- Start points in generated code!
- End points in framework code!
- ⇒ No way to use standard Java.
- ⇒ Only AOP / AspectJ applies.



Why use load-time weaving?

- Recompiling of framework code not allowed
 - ◆ But end-points of measurement in framework code
 - ◆ Load-time weaving easy to use for all kinds of bundles
- Compile-time weaving not mature in headless build!
 - ◆ Eclipse 3.2.0
 - ◆ AJDT patch for PDE build since AJDT 1.4.1 (for Eclipse 3.2.1)
 - ◆ We tried, but ...



For more info on this success story ...

... read the article
“Performance-Logging mit AspectJ”
in the upcoming Eclipse Magazin (Vol. 15).



What's next for Equinox Aspects?

- “Ease of Use”
 - ◆ Quick start tutorial
 - ◆ More examples
 - ◆ Maybe cheat sheets
- Advanced tooling
 - ◆ Bring AJDT navigation and debugging features to OSGi
- More dynamics
- Improved caching
- Graduation... 😊



Conclusions

- Equinox Aspects brings full AOP to OSGi
 - ◆ Via integrated load-time weaving
 - ◆ Combines OSGi and AOP modularity features
- Can be used for production systems today

- Give it a try
<http://www.eclipse.org/equinox/incubator>



Thank you for your attention!

Q&A

Heiko Seeberger: heiko.seeberger@metafinanz.de

Martin Lippert: martin.lippert@akquinet.de