Java - Cooler, more Productive & Effective Than You May Think

[kill the crazyness]

blog.adam-bien.com / twitter:@AdamBien

- Expert Group Member (jcp.org) of Java EE 6, Java EE 7, JPA 2.1, EJB 3.2, CDI 1.1, JMS 2.0 (...)

- Java Champion, (JavaONE) speaker + rockstar, freelancer, consultant and author: >100 articles, 7 German books,

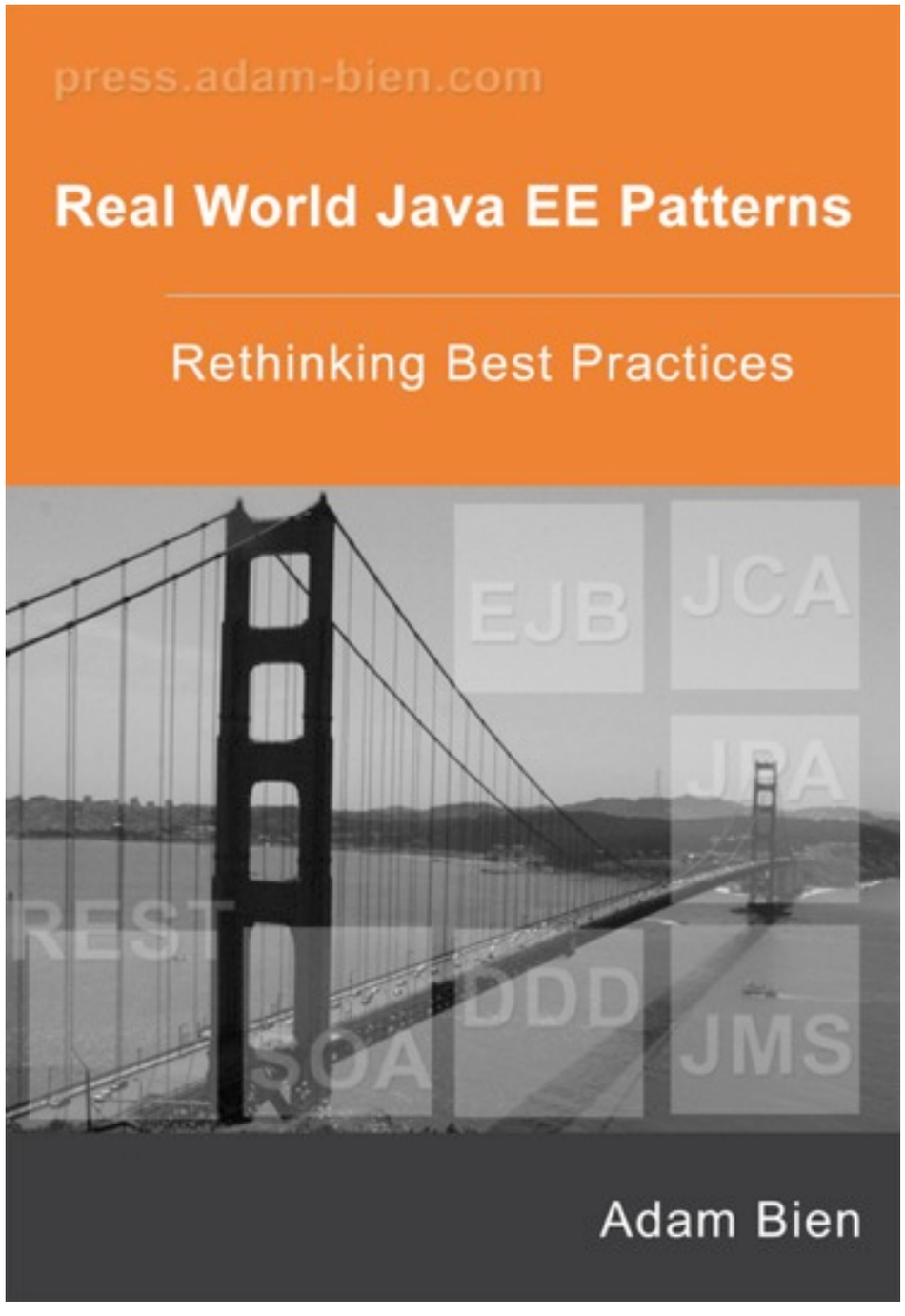
- Author: “Real World Java EE Patterns– Rethinking Best Practices” and “Real World Java EE Night Hacks” http://press.adam-bien.com

- NEW: **workshops**.adam-bien.com

- http://kenai.com/projects/javaee-patterns/

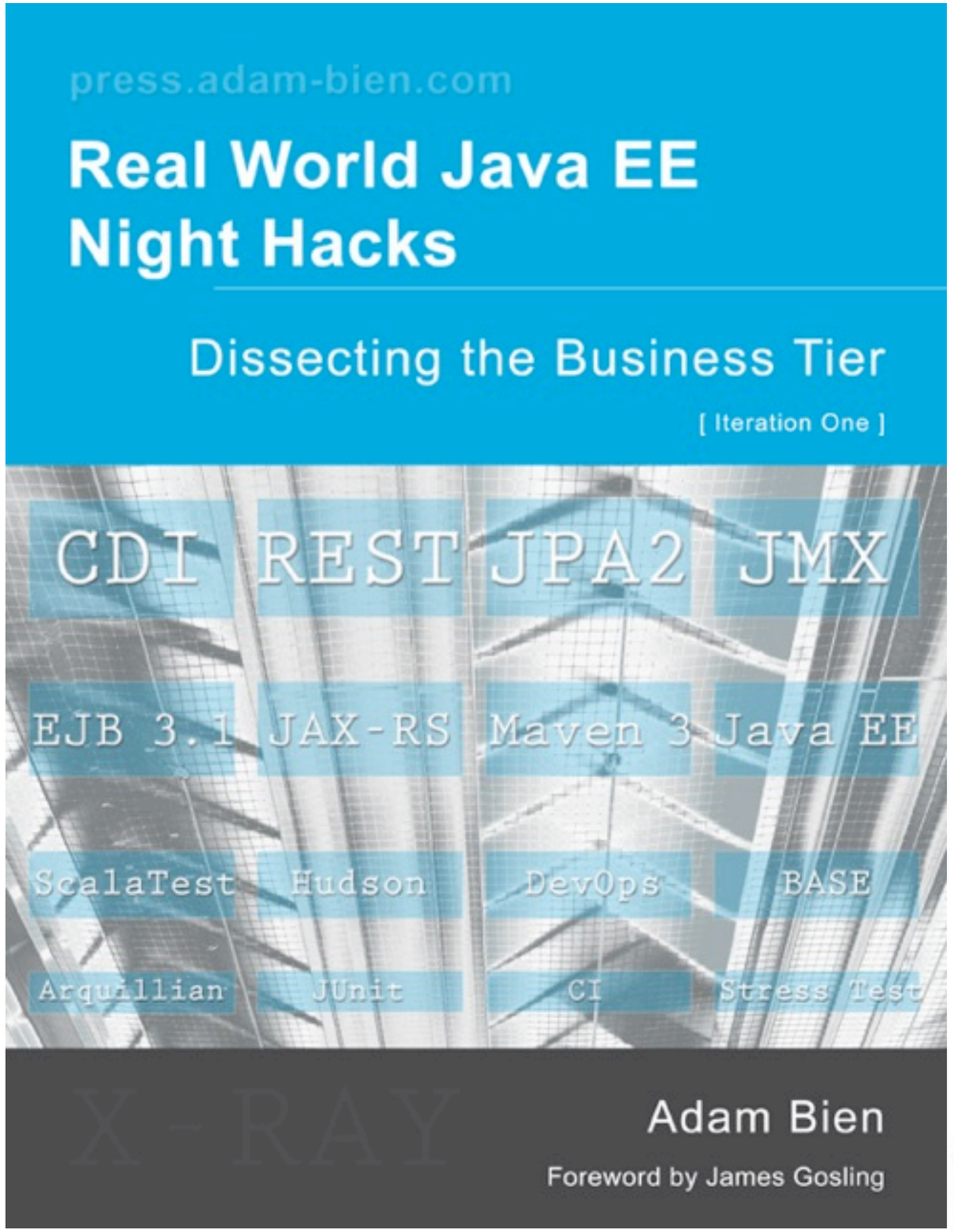
  http://java.net/projects/x-ray

blog.adam-bien.com / twitter:@AdamBien



Adam Bien,
press.adam-bien.com

blog.adam-bien.com / twitter:@AdamBien



Adam Bien,
press.adam-bien.com

blog.adam-bien.com / twitter:@AdamBien

0xcafebabe

Is Java the New Cobol?

(hopefully)

Twitter, Azul are working on openJDK

IDEs, Tools and Libraries

Easy To Learn

Not Many Surprises

VooDoo Is Hard To Achieve (Boring)

There Are No "Famous One Liners"

Really Hard To Obfuscate (Commercial Tools are Available)

There Are No "International Obfuscated Java Code Contest" (www.ioccc.org)

Performance

Startup

JMX

```
MBeanServer mbs  =
ManagementFactory.getPlatformMBeanServer();

String
compositeName=“hello:type=com.abien.hello”;

ObjectName mbeanName= new
ObjectName(compositeName);

this.mbs.registerMBean(mbean,mbeanName);
```

Aspects (Dynamic Proxy)

...since JDK 1.3

```
public class TracingInvocationHandler implements InvocationHandler{

public Object invoke(Object proxy, Method m, Object[] args) throws Throwable {

        result = m.invoke(obj, args);

} catch (InvocationTargetException e) {

        throw e.getTargetException();

} catch (Exception e) {

//infrastructure exception

} finally {

}

return result;

}
```

```
Proxy.newProxyInstance(

classLoader,

allInterfaces,

handler);
```

Plugins (ServiceLoader)

java.util.ServiceLoader

- Searches for implementations for an interface (from meta-inf/services).
- Instantiates lazily the realization and caches the products.
- Uses generics – is convenient to use.
- NetBeans RCP is based on it...

```
ServiceLoader<S> loader =
ServiceLoader.load(clazz);

 for (Iterator<S> it = loader.iterator();
it.hasNext();) {

        ... it.next();
```

```
HelloService service =
LoaderUtility.load(HelloService.class);
```

Content of the configuration file:

com.abien.HelloWorldService

```
ServiceLoader<Compressor> loader =
load(Compressor.class);

for (Iterator<Compressor> it = loader.iterator();
it.hasNext();) {
    S impl = it.next();
```

RMI

```
public interface HelloWorldService extends
Remote{
public String helloFromServer(String
greeting) throws RemoteException;
}
```

```
public class HelloWorldServant extends
UnicastRemoteObject implements HelloWorldService{

protected HelloWorldServant() throws
RemoteException {
        super();
    }
    public String helloFromServer(String greeting) {
        return greeting + " time: " + new Date();
    }
}
```

```
HelloWorldService h = (HelloWorldService)
Naming.lookup("rmi://localhost:1200/hugo");
```

```
String msg  = h.helloFromServer("hello ");
```

XML Processing

XML Processing

- Stax
- JAXB
- XMLEncoder / XMLDecoder

```
XMLEncoder e = new XMLEncoder(new
FileOutputStream("Test.xml"));

e.writeObject(new JButton("Hello, world"));
e.close();
```

Scripting

```
ScriptEngineManager se = new ScriptEngineManager();

        se  = engineManager.getEngineByName("JavaScript");

        Object retVal = null;

        Bindings bindings = new SimpleBindings();

        bindings.put(“digit", 21);

        retVal = se.eval(21*2,bindings);

        return (Double) retVal;
```

DerbyDB

Logging

Concurrency

Java FX 2

Jigsaw, Lambda Expressions

New NIO

WORKS IN JDK 1.6!

//Z means: "The end of the input but for the final terminator, if any"

```
String output = new Scanner(new
File("file.txt")).useDelimiter("\
\Z").next();
        System.out.println("" + output);
```

```
import static java.nio.charset.Charset.*;
import static java.nio.file.Files.*;
import static java.nio.file.Paths.*;

public class FileReader {

    public static void read(String file) throws IOException{
        List<String> allLines = readAllLines(get(file), defaultCharset());
        for (String line : allLines) {
            System.out.println(line);
        }
    }

    public static void main(String[] args) throws IOException {
        read("./readme.txt");
    }
}
```

```
import java.io.BufferedReader;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

        Path file = Paths.get("./readme.txt");
        BufferedReader reader =
    Files.newBufferedReader(file, Charset.defaultCharset());
        StringBuilder content = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            content.append(line).append("/n");
        }
        System.out.println("Content: " + content.toString());
```

Fluenza

```
import static org.mockito.Mockito.*;

    RentalService cut;

    @Before

    public void initDI(){

        cut = new RentalService();

        cut.cv = mock(CustomerValidation.class);

}

@Test(expected=IllegalArgumentException.class)

public void buy() {

    when(cut.cv.cashAmount(anyInt())).thenReturn(false);

    cut.buy("tesla");
```

```
training.login(“adam”,”bien”).
        running().
        today().
        averagePulse(135).
        comment("ok").
        duration(0,65,1).
        maxPulse(165).
        distance(12).
        wasRace().
        weather(SUN).
        add();
```

With A Little Help Of EE

[GoF] Observer

```
@Inject Event<Vehicle> events;

events.fire(new Vehicle());

//-------

void onReportedVehicle(@Observes Vehicle
vehicle){ }
```

Scheduler

```
@Schedule(hour="*",minute="*",second="*/5")
public void flushCache(){}
```

Asynchronous Processing

```
@Asynchronous
public Future<String> assemble(Vehicle v){}
```

[GoF] Factory

```
@Produces
public String getConfiguration() {}
```

blog.adam-bien.com / twitter:@AdamBien

[GoF] Decorator

```
@AroundInvoke

public Object audit(InvocationContext ic)
throws Exception{
    Method method = ic.getMethod();
    System.out.println("AUDIT: " + method);
    return ic.proceed();
}
```

[GoF] Typesafe Decorator

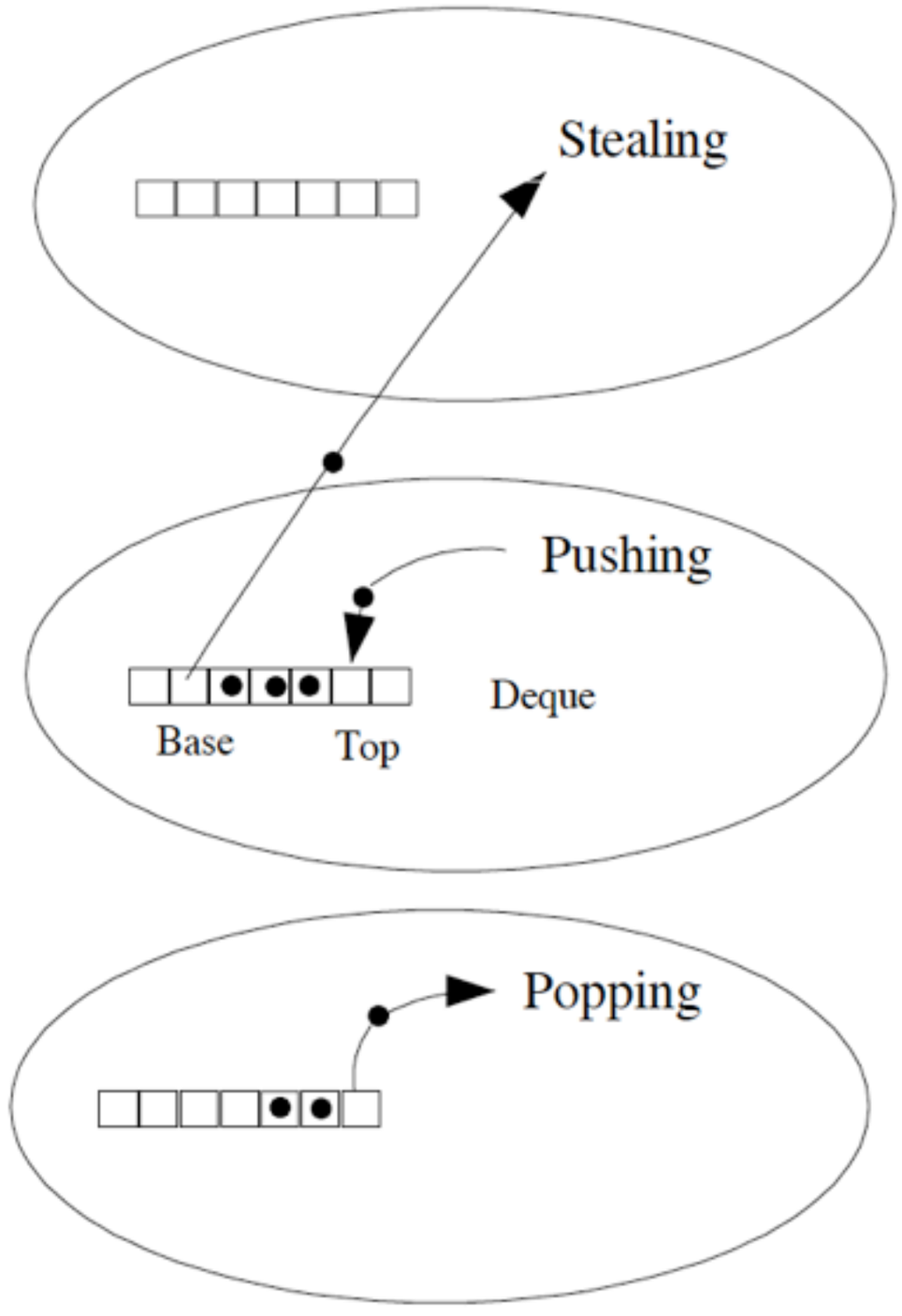
```
public class NiceMessenger implements
Messenger {

    @Override
    public String morning(){
        return "morning";
    }
}
```

[GoF] Typesafe Decorator

```
@Decorator
public abstract class NicenessExtender implements Messenger{
    @Inject @Delegate
    private Messenger messenger;
    @Override
    public String morning() {
        return "Good " + messenger.morning();
    }
}
```

ForkJoin

```
Result solve(Problem problem) {
if (problem is small)
        directly solve problem
else {
        split problem into independent parts fork new
        subtasks to solve each part join all subtasks
        compose result from subresults
}
```



Stolen from Doug's Lea Paper :-)

blog.adam-bien.com / twitter:@AdamBien

Thank You!

blog.adam-bien.com
twitter.com/AdamBien