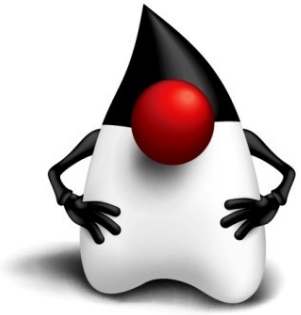
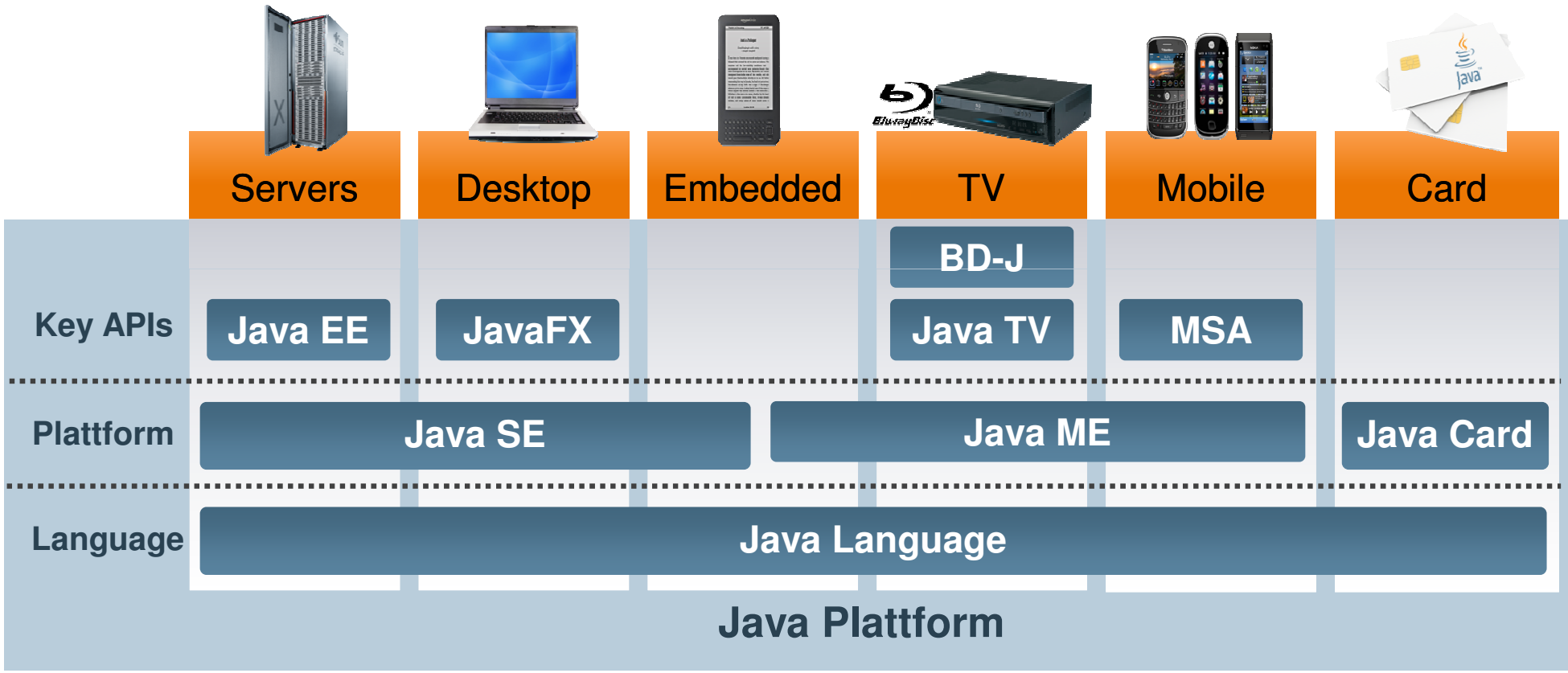
Lucky Seven

Die Java Plattform Strategie

Wolfgang Weigend
Sen. Leitender Systemberater
Java Technologie und Architektur



Die Java Plattform



ORACLE

Was die Zukunft bringt

- Java Strategie Treiber
- Java Community
- JDK 7 Review
- JDK 8 und darüber hinaus
- Entwicklungsumgebung
- JavaFX & Java ME
- Java EE

Java Strategie

- **Höchste Priorität für Java!**
  - Zunehmendes Investment in die Plattform
  - Kontinuierliche Bewegung in Richtung ”free & open” (OpenJDK)
  - Verbesserter Support für Entwickler Community, Open Source Community und JCP
- **Niedrigere Priorität: Direkter Umsatz mit Java**
  - Support für Oracle Software und Hardware
  - Mehrwert auf Basis der Java Plattform, Enterprise Support

Bisher ausgelieferte Technologie

	Verfügbarkeit
Java SE	☑ **JDK 7 for Mac OS X Developer Preview** ☑ JDK 7 ☑ JRockit released gratis under same license as JDK
Java EE	☑ GlassFish 3.1.1 ☑ Java EE 7 specification development underway
Java FX	☑ Java FX 2.0 ☑ Java FX 2.0 for Mac OS X Developer Preview
Java ME/ Embedded	☑ Oracle Java Wireless Client 3.0 ☑ Oracle Java Embedded Client 1.0 ☑ Java SE for Embedded 7 ☑ Java ME SDK 3.0.5 and LWUIT 1.5

ORACLE

Einige Pläne wurden bereits verwirklicht und neue Projekte gestartet

JDK 7

- Available for Windows, Linux, Solaris, embedded platforms
- Mac OS X Developer Preview released

JVM Konvergenz

- Initial JRockit/HotSpot Konvergenz in JDK 7
- JRockit released under ”gratis” JDK license (BCL)

Open Source

- OpenJDK official Java SE 7 Reference Implementation
- OpenJDK project initiated for JDK 8

JVM Strategie: HotSpot und JRockit Konvergenz

- JRockit und HotSpot werden in einem mehrjährigen Prozess zu einer einheitlichen JVM verschmolzen:
  "HotRockit"
  - unter Berücksichtigung der besten Funktionsmerkmale beider JVM´s
- Die Arbeitsergebnisse wurden inkrementell zum OpenJDK zugeführt
  - JRockit Performance Funktionalität - Bereits im JDK 7
- Ziele
  - Maximale Performance und Skalierbarkeit
  - Multi-Core und Parallelisierungs Optimierungen
  - Feingranulares Monitoring, Profiling und Management
  - Extrem effiziente Garbage Collection

HotRockit Converged JVM (1)

JCMD Command Line utility to enumerate and send commands to running JVMs - ***JDK 7 Time Frame***

JMX Agent Update - ***JDK 7 Time Frame***

Java Discovery Protocol JDP - ***JDK 7 Update Time Frame***

- Multicasting heartbeat for JVM services
- Used to discover manageable JVMs on the network
- Also to discover JVM's no longer running
- Normally used with the JMX management agent

MBean Updates - ***JDK 7 Time Frame***

- Many MBeans from JRockit's JMXMAPI Ported
- Means better support for Hotspot in the Mission Control Console
- Examples:

```
getJVMGeneratedCPULoad();    getAllocatedBytes(long[] threadIDs)
getInvocationCount(MethodID id);      getTiming(MethodID id)
DiagnosticCommandMBean;               PerfCounterMBean
```


HotRockit Converged JVM (2)

No More Perm-Gen - *JDK 7 Time Frame*

- Perm-gen will be removed
- Will use native memory and allocate as needed
- No need to decide the required size up front
- No need for tuning

Java Flight Recorder - *Update to JDK 7*

- Always on
- Very low overhead
- Dump data anytime
- Go back in time to see what lead up to a problem

Memleak Server - *JDK 8 Time Frame*

- Low overhead memory analyzer
- In-situ analysis

Other Improvements - *After JDK 8 Time Frame*

- Deterministic GC (Soft real-time GC; Pause time target)
- Compiler optimizations
- Smaller object headers

Evolutionäre Weiterenwicklung der Sprache

- **Langzeit Perspektive (20+ Jahre)**
  - Ideen reifen in ausgeprägten experimentellen Sprachen
  - Abwärtskompatibilität bleibt extrem wichtig für die Sprache
- **Periodische Anreicherung von ausgewählter Funktionalität**
  - Verbesserte Entwicklerproduktivität
  - Bewahrt Klarheit und Einfachheit

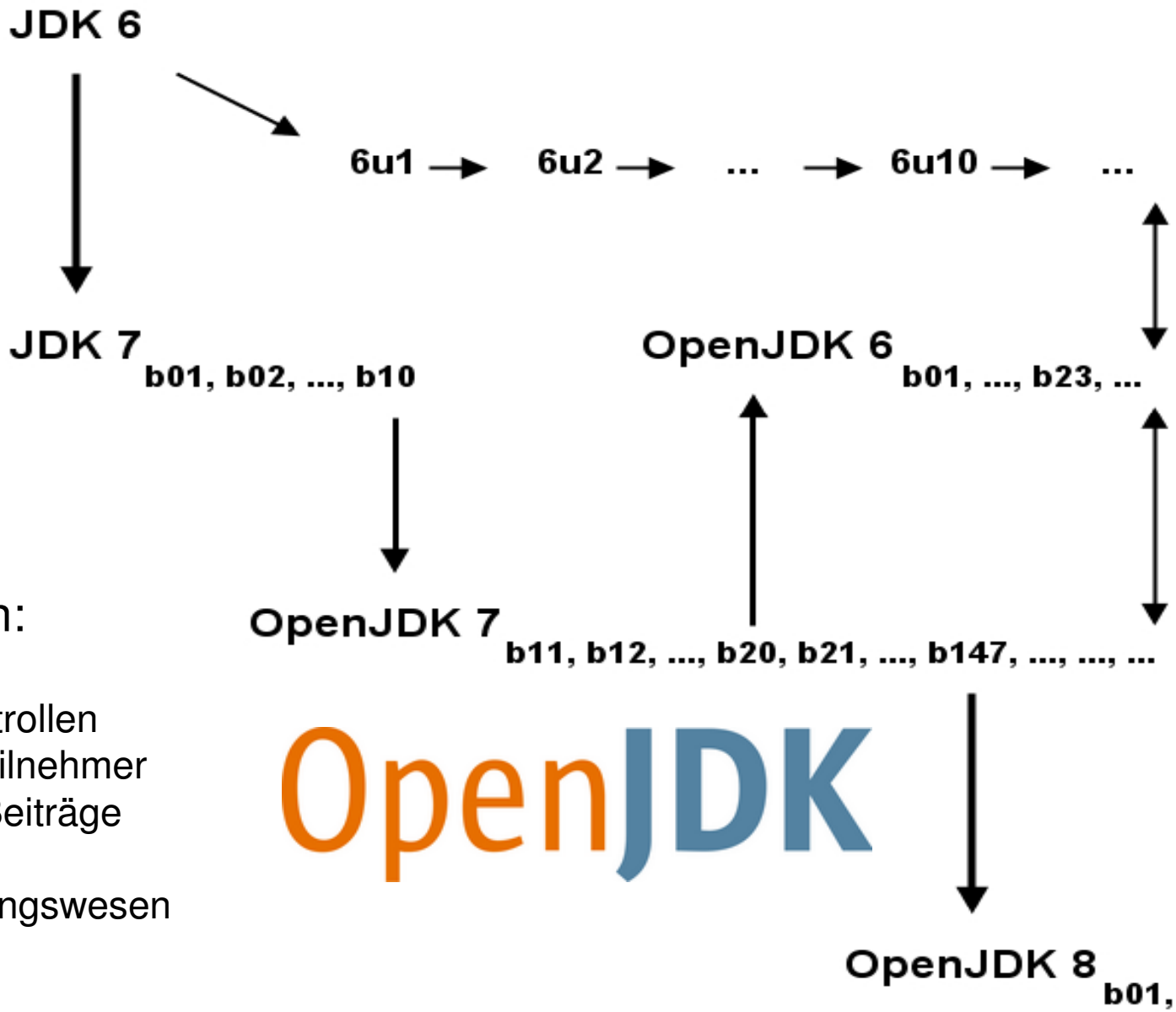
Teilnahme erwünscht!

OpenJDK

- Oracle verpflichtet sich zur besten Open-Source Java Implementation
- Ziel der neuen OpenJDK Richtlinien
  - Fördern vom langfristigen Bestand & Wachstum der Community
  - Basis für offene, transparente und leistungsbezogene Aktivitäten der Mitglieder
- Neue OpenJDK Releases
  - Liste der Features unter openjdk.java.net/projects/jdk7/features
- Software-Hersteller beteiligen sich:



OpenJDK Stammbaum

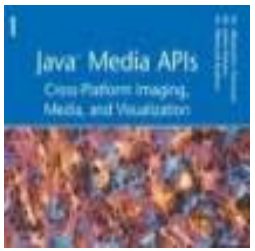


OpenJDK Richtlinien:

- Rollen & Gruppen
- Gruppenrollen & Projektrollen
- Projekte & OpenJDK Teilnehmer
  - Bug-Fixing & Code-Beiträge
- Governing Board
- Reporting & Entscheidungswesen
- Abstimmung

JCP wird reformiert: JSR-348

- Mehr Entwickler im Executive Committee
  - SOUJava
  - Goldman Sachs
  - London JavaCommunity
- JCP startet ein Programm zur Reform
  - JSR 348: Towards a new version of the JCP



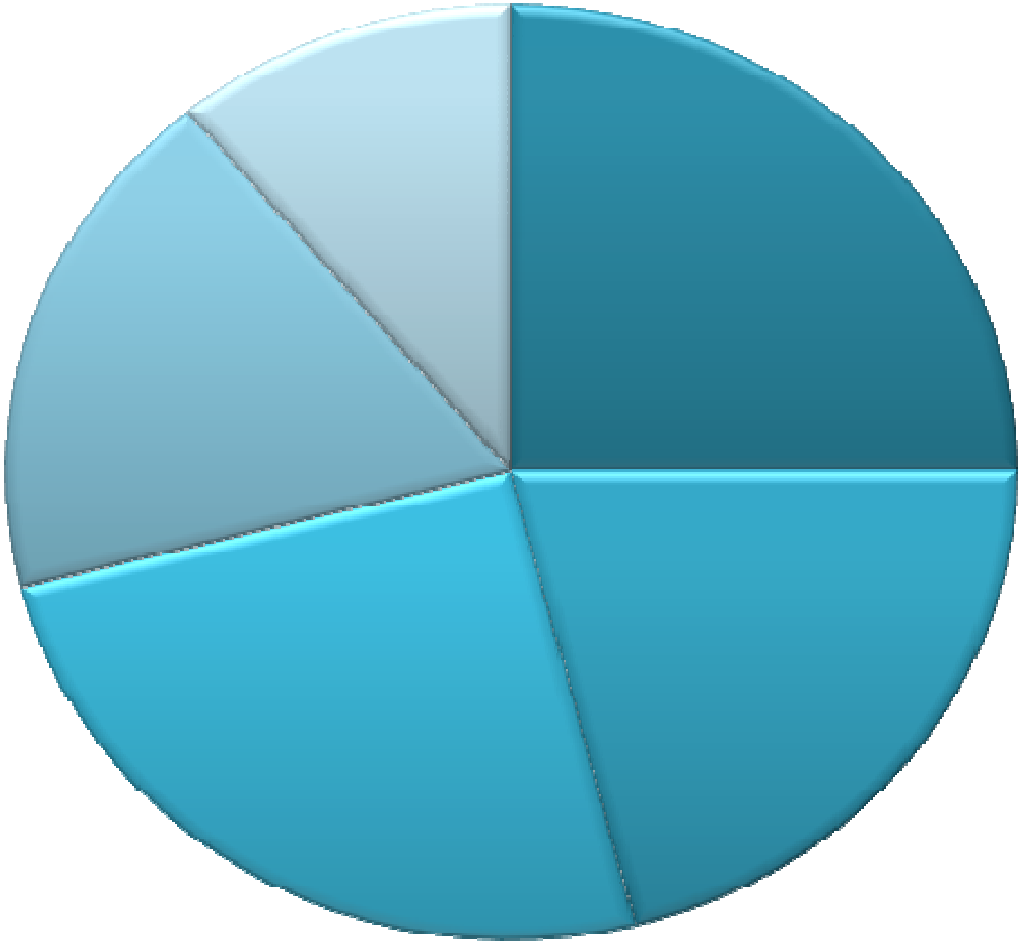
ORACLE

Zusammenarbeit mit der Community

	Aktivitäten
OpenJDK	☑ IBM, Apple, SAP, Azul join OpenJDK ☑ OpenJDK community bylaws ratified ☑ OpenJDK becomes official Java SE 7 Reference Implementation ☑ JVM Language Summit held July 2011 ☑ OpenJDK Twitter feed launched with thousands of followers
JCP	☑ JSR for Java SE 7 passed & JSR for Java SE 8 submitted & in process ☑ SouJava, London JUG elected to JCP Executive Committee ☑ JSR 348 in process for greater transparency ☑ JSR 342 submitted for Java EE 7 platform
JUGs	☑ Oracle lead bi-weekly JUG leaders call ☑ Oracle lead JUG leader summit January 2011 ☑ Over 250 JUGs from 50+ countries involved in JDK 7 launch

Wird Java 7 eingesetzt?

Umfrage auf java.net: "*Have you tried out Java 7 yet?*"



- Yes, and I`m working with it regulary (25%)
- I`ve experimented with it a bit (21%)
- I plan to get started with Java 7 soon (25%)
- I`m waiting for a bug fix release (18%)
- No, and I don´t plan to (11%)

JDK 7 Update Releases

Fehlerbereinigung

OpenJDK

- Projekt entwickelt Updates für JDK 7
- Sponsor ist die Build Group
- Mailing Liste lautet jdk7u-dev
  - Technische Diskussion zum JDK 7 Updates Projekt
  - Archive vorhanden
  - Nachrichten an alle Teilnehmer verschicken:
    - jdk7u-dev@openjdk.java.net
  - jdk7u-dev Subscription
- **Java SE 7 Update 2 Developer Preview Releases**
  - **Liste der Änderungen im aktuellen JDK 7u2 build b11**
  - **Projekt Feedback Forum für Java SE Snapshots**
  - **Report a Bug oder Request a Feature**

ORACLE

JDK 7 Funktionsumfang

- Coin – Kleine Sprachverbesserungen (JSR 334)
- Dynamic Language Support (JSR 292)
- Concurrency und Collections Updates (JSR 166y)
- Netzwerk und File System (JSR 203)
- Sicherheit
- Internationalisierung
- Weitere Verbesserungen
- JVM Konvergenz

7

ORACLE

Strings in Switch Statements

```
int monthNameToDays(String s, int year) {
  switch(s) {
    case "April": case "June":
    case "September": case "November":
      return 30;

    case "January": case "March":
    case "May": case "July":
    case "August": case "December":
      return 31;

    case "February”:
      ...
    default:
      ...
```


Diamond Operator

- Pre-generics

```
List strList = new ArrayList();
```

- With Generics

```
List<String> strList = new ArrayList<String>();
List<Map<String, List<String>> strList =
  new ArrayList<Map<String, List<String>>>();
```

- With diamond (<>) compiler infers type

```
List<String> strList = new ArrayList<>();
List<Map<String, List<String>> strList =
  new ArrayList<>();
```


Copying a File

```
InputStream in = new FileInputStream(src);
OutputStream out = new FileOutputStream(dest);

byte[] buf = new byte[8192];
int n;

while (n = in.read(buf)) >= 0)
  out.write(buf, 0, n);
```


Copying a File (Better, but wrong)

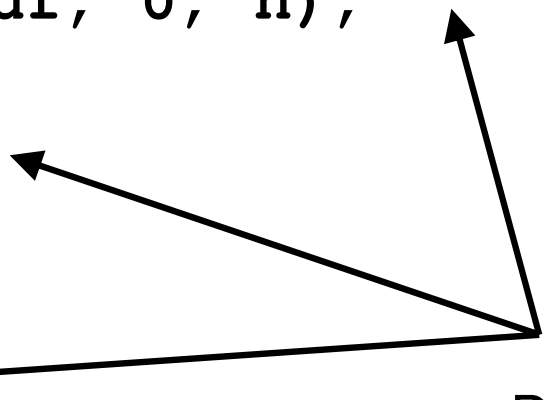
```
InputStream in = new FileInputStream(src);
OutputStream out = new FileOutputStream(dest);

try {
  byte[] buf = new byte[8192];
  int n;
  while (n = in.read(buf)) >= 0)
    out.write(buf, 0, n);
} finally {
  in.close();
  out.close();
}
```


Copying a File (Correct, but complex)

```
InputStream in = new FileInputStream(src);
try {
  OutputStream out = new FileOutputStream(dest);
  try {
    byte[] buf = new byte[8192];
    int n;
    while (n = in.read(buf)) >= 0)
      out.write(buf, 0, n);
  } finally {
    out.close();
  }
} finally {
  in.close();
}
```


Copying a File (Correct, but complex)

```
InputStream in = new FileInputStream(src);
try {
  OutputStream out = new FileOutputStream(dest);
  try {
    byte[] buf = new byte[8192];
    int n;
    while (n = in.read(buf)) >= 0)
      out.write(buf, 0, n);
  } finally {
    out.close();
  }
} finally {
  in.close();
}
```



Exception thrown from potentially three places. Details of first two could be lost

 |

Automatic Resource Management

```
try (InputStream in = new FileInputStream(src),
     OutputStream out = new FileOutputStream(dest))
{
  byte[] buf = new byte[8192];
  int n;
  while (n = in.read(buf)) >= 0)
    out.write(buf, 0, n);
}
```


The Details

- Compiler de-sugars try-with-resources into nested try-finally blocks with variables to track exception state
- Suppressed exceptions are recorded for posterity using a new facility of Throwable
- API support in JDK 7
  - New superinterface **`java.lang.AutoCloseable`**
  - All **`AutoCloseable`** and by extension java.io.Closeable types useable with try-with-resources
  - Anything with a **`void close()`** method is a candidate
  - JDBC 4.1 retro-fitted as **`AutoCloseable`** too

More Informative Backtraces

```
java.io.IOException
        at Suppress.write(Suppress.java:19)
        at Suppress.main(Suppress.java:8)
        Suppressed:  java.io.IOException
            at Suppress.close(Suppress.java:24)
            at Suppress.main(Suppress.java:9)
        Suppressed:  java.io.IOException
            at  Suppress.close(Suppress.java:24)
            at  Suppress.main(Suppress.java:9)
```


Varargs Warnings

```
class Test {
   public static void main(String... args) {
      List<List<String>> monthsInTwoLanguages =
        Arrays.asList(Arrays.asList("January",
                                    "February"),
                      Arrays.asList("Enero",
                                    "Febrero" ));
   }
}
```

```
Test.java:7:  warning:
[unchecked]  unchecked  generic  array  creation
for  varargs parameter  of  type  List<String>[]
         Arrays.asList(Arrays.asList("January",
                       ^
1  warning
```


Varargs Warnings Revised

- New mandatory compiler warning at suspect varargs method declarations
- By applying an annotation at the declaration, warnings at the declaration *and call sites* can be suppressed
- **`@SuppressWarnings(value = "unchecked")`**
- **`@SafeVarargs`**

Lots of Exceptions

```
try {
...
} catch(ClassNotFoundException cnfe) {
   doSomethingClever(cnfe);
   throw cnfe;
} catch(InstantiationException ie) {
   log(ie);
   throw ie;
} catch(NoSuchMethodException nsme) {
   log(nsme);
   throw nsme;
} catch(InvocationTargetException ite) {
   log(ite);
   throw ite;
}
```


Multi-Catch

```
try {
...
} catch (ClassCastException e) {
  doSomethingClever(e);
  throw e;
} catch(InstantiationException |
  NoSuchMethodException |
  InvocationTargetException e) {
log(e);
throw e;
}
```

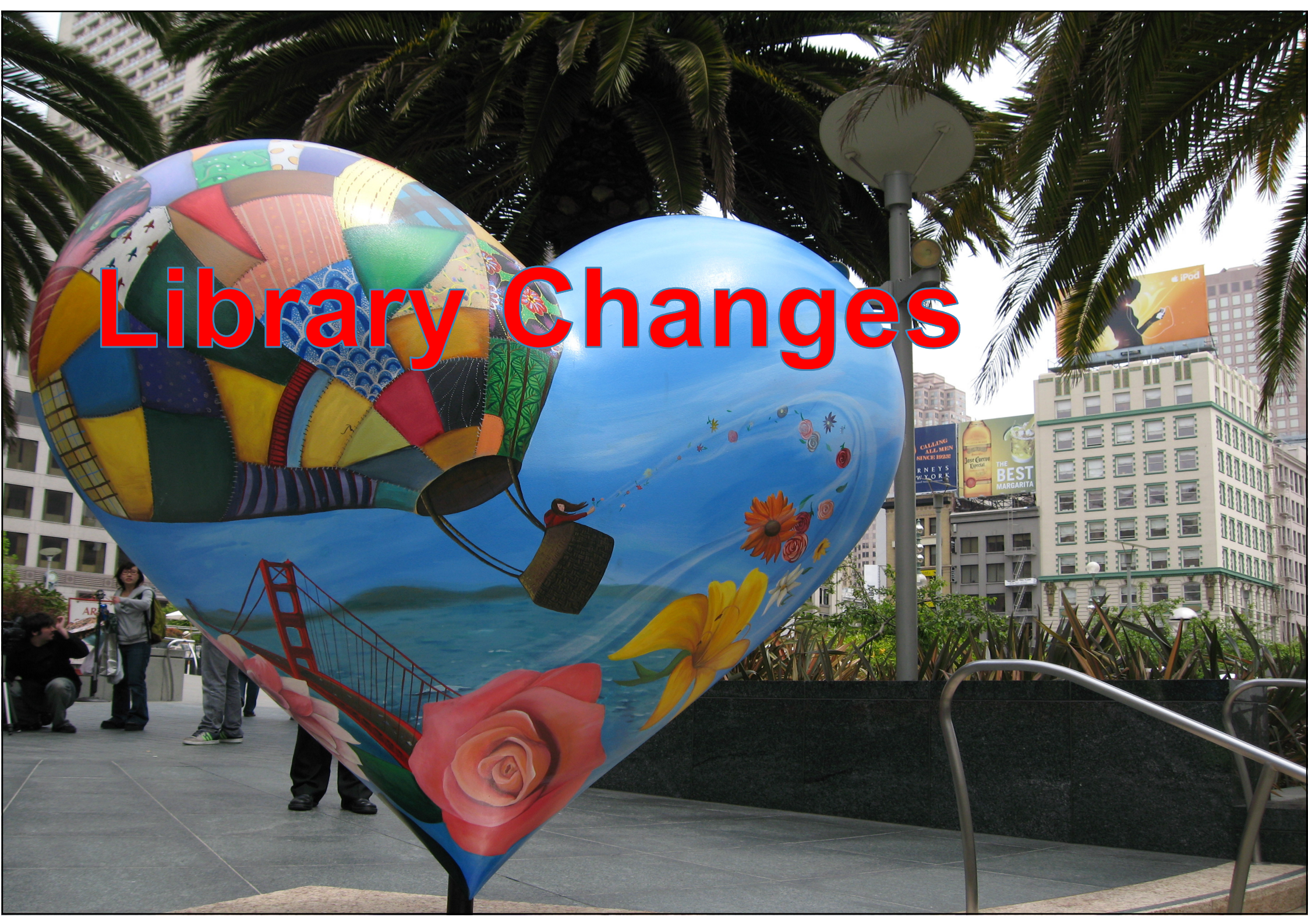

IDE Support

- Beta support in Eclipse
  - http://thecoderlounge.blogspot.com/2011/06/java-7-support-in-eclipse-jdt-beta.html
- IntelliJIDEA 10.5
- NetBeans7.0
  - http://netbeans.org/kb/docs/java/javase-jdk7.html
- Demo

```
37
38
39
40
42
43
44
45
```

Can be replaced with multicatch
--
(Alt-Enter shows hints)

```
            NotFoundException("adasdf");
        } catch (FileNotFoundException fnfo) {
            fnfo.printStackTrace();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
```



Library Changes
CALLING ALL MEN SINCE 1923!
THE BEST MARGARITA
iPod

New I/O 2 (NIO2) Libraries

JSR 203

- Original Java I/O APIs presented challenges for developers
- Need something better than java.io.File
  - Doesn't work consistently across platforms
  - No useful exceptions when a file operation fails
  - Missing basic operations (file copy, move, ...)
  - Limited support for symbolic links
  - Limited support for file attributes, performance issues
  - No way to plug-in other file system implementations
- Java NIO2 solves these problems

Java NIO2 Features

- Path is a replacement for File
  - Biggest impact on developers
- Better directory support
- Files
  - Static methods to operate on files and directories
  - Support for symbolic links
- FileStore
  - Represents underlying file storage (partition, concrete file system)
- FileSystem
  - SPI interface to a filesystem (FAT, ZFS, Zip archive, network, etc)
- Access to file metadata

`Path` Class

- Equivalent of **`java.io.File`** in the new API
  - Immutable
- Have methods to access and manipulate **`Path`**
- Supports old libraries
  - Create File from Path using toFile

```
//Make a reference to the path
Path home = Paths.get("/home/fred");

//Resolve tmp from /home/fred -> /home/fred/tmp
Path tmpPath = home.resolve("tmp");

//Create a relative path from tmp -> ..
Path relativePath = tmpPath.relativize(home)

File file = relativePath.toFile();
```


File Operation – Copy, Move

- File copy is really easy
  - With fine grain control

```
Path src = Paths.get("/home/fred/readme.txt");
Path dst = Paths.get("/home/fred/copy_readme.txt");

Files.copy(src, dst,
   StandardCopyOption.COPY_ATTRIBUTES,
   StandardCopyOption.REPLACE_EXISTING);
```

- File move is supported
  - Optional atomic move supported

```
Path src = Paths.get("/home/fred/readme.txt");
Path dst = Paths.get("/home/fred/readme.1st");

Files.move(src, dst, StandardCopyOption.ATOMIC_MOVE);
```


Directories

- DirectoryStream iterate over entries
  - Scales to large directories
  - Uses less resources
  - Smooth out response time for remote file systems
  - Implements **`Iterable`** and **`Closeable`** for productivity
- Filtering support
  - Build-in support for glob, regex and custom filters

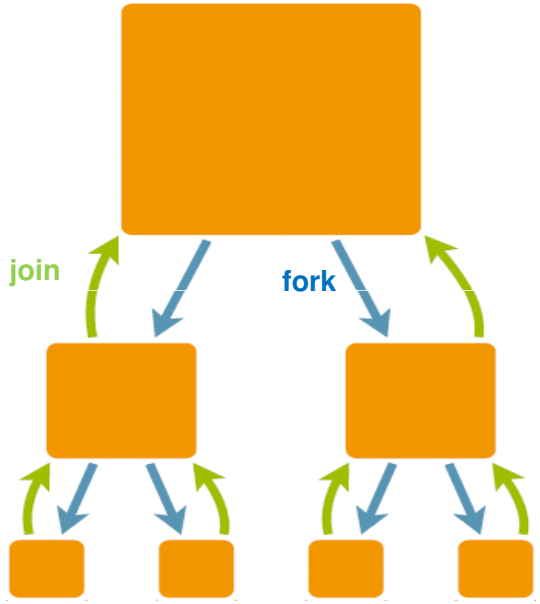
```
Path srcPath = Paths.get("/home/fred/src");

try (DirectoryStream<Path> dir =
   srcPath.newDirectoryStream("*.java")) {
for (Path file: dir)
   System.out.println(file.getName());
}
```


Concurrency APIs

- JSR166y
  - Update to JSR166x which was an update to JSR166
- Adds a lightweight task framework
  - Also referred to as Fork/Join
- **`Phaser`**
  - Barrier similar to **`CyclicBarrier`** and **`CountDownLatch`**
- **`TransferQueue`** interface
  - Extension to **`BlockingQueue`**
  - Implemented by **`LinkedTransferQueue`**

Fork Join Framework

- Goal is to take advantage of multiple processor
- Designed for task that can be broken down into smaller pieces
  - Eg. Fibonacci number fib(10) = fib(9) + fib(8)
- Typical algorithm that uses fork join




```
if I can manage the task
perform the task
else
fork task into x number of smaller/similar task
join the results
```


Key Classes

- **`ForkJoinPool`**
  - Executor service for running **`ForkJoinTask`**
- **`ForkJoinTask`**
  - The base class for forkjoin task
- **`RecursiveAction`**
  - A subclass of **`ForkJoinTask`**
  - A recursive resultless task
  - Implements **`compute()`** abstract method to perform calculation
- **`RecursiveTask`**
  - Similar to **`RecursiveAction`** but returns a result

ForkJoin Example – Fibonacci

```
public class Fibonacci extends RecursiveTask<Integer> {
    private final int number;
    public Fibonacci(int n) { number = n; }

    @Override protected Integer compute() {
        switch (number) {
            case 0: return (0);
            case 1: return (1);
            default:
              Fibonacci f1 = new Fibonacci(number - 1);
              Fibonacci f2 = new Fibonacci(number - 2);
                f1.fork(); f2.fork();
                return (f1.join() + f2.join());
        }
    }
}
```

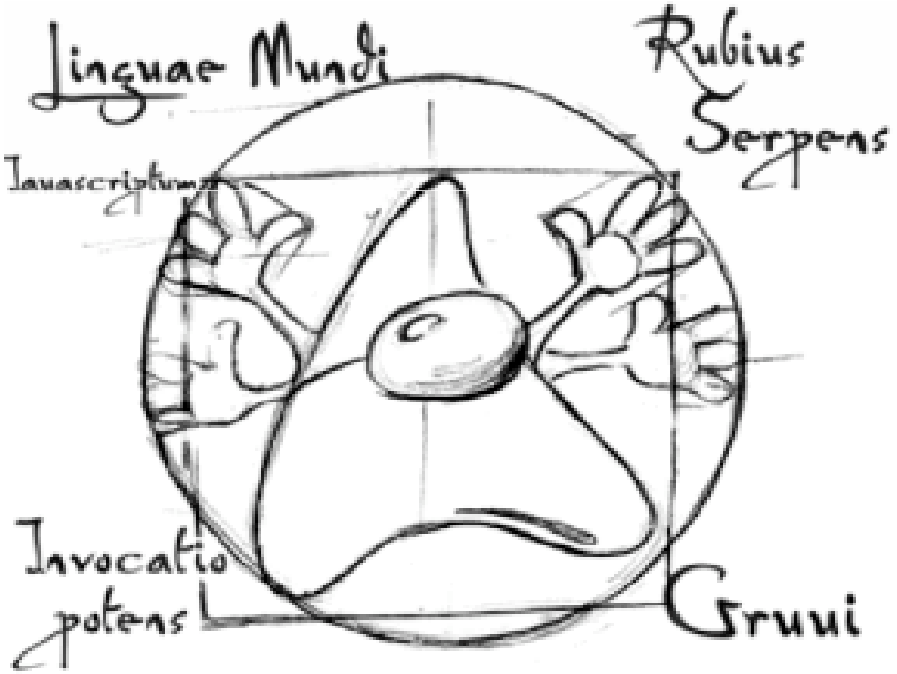

ForkJoin Example – Fibonacci

```
ForkJoinPool pool = new ForkJoinPool();
Fibonacci r = new Fibonacci(10);
pool.submit(r);

while (!r.isDone()) {
//Do some work
...
}

System.out.println("Result of fib(10) = "
   + r.get());
```


ForkJoin Performance Discussion

- Choosing the sequential threshold
  - Smaller tasks increase parallelism
  - Larger tasks reduce coordination overhead
  - Ultimately you must profile your code
- Minimizes overhead for compute-intensive tasks
  - Not recommended for tasks that mix CPU and I/O activity
- A portable way to express many parallel algorithms
  - Reasonably efficient for a wide range of core counts
  - Library-managed parallelism

The DaVinci Machine Project (JSR-292)
(A multi-language renaissance for the JVM)



Languages Like Virtual Machines

- Programming languages need runtime support
  - Memory management / Garbage collection
  - Concurrency control
  - Security
  - Reflection
  - Debugging integration
  - Standard libraries
- Compiler writers have to build these from scratch
- Targeting a VM allows reuse of infrastructure

JVM Specification

"The Java virtual machine knows nothing about the Java programming language, only of a particular binary format, the class file format."

1.2 The Java Virtual Machine Spec.

Languages Running on the JVM

Groovy
JRuby
…
…
Scala
Clojure

InvokeDynamic Bytecode

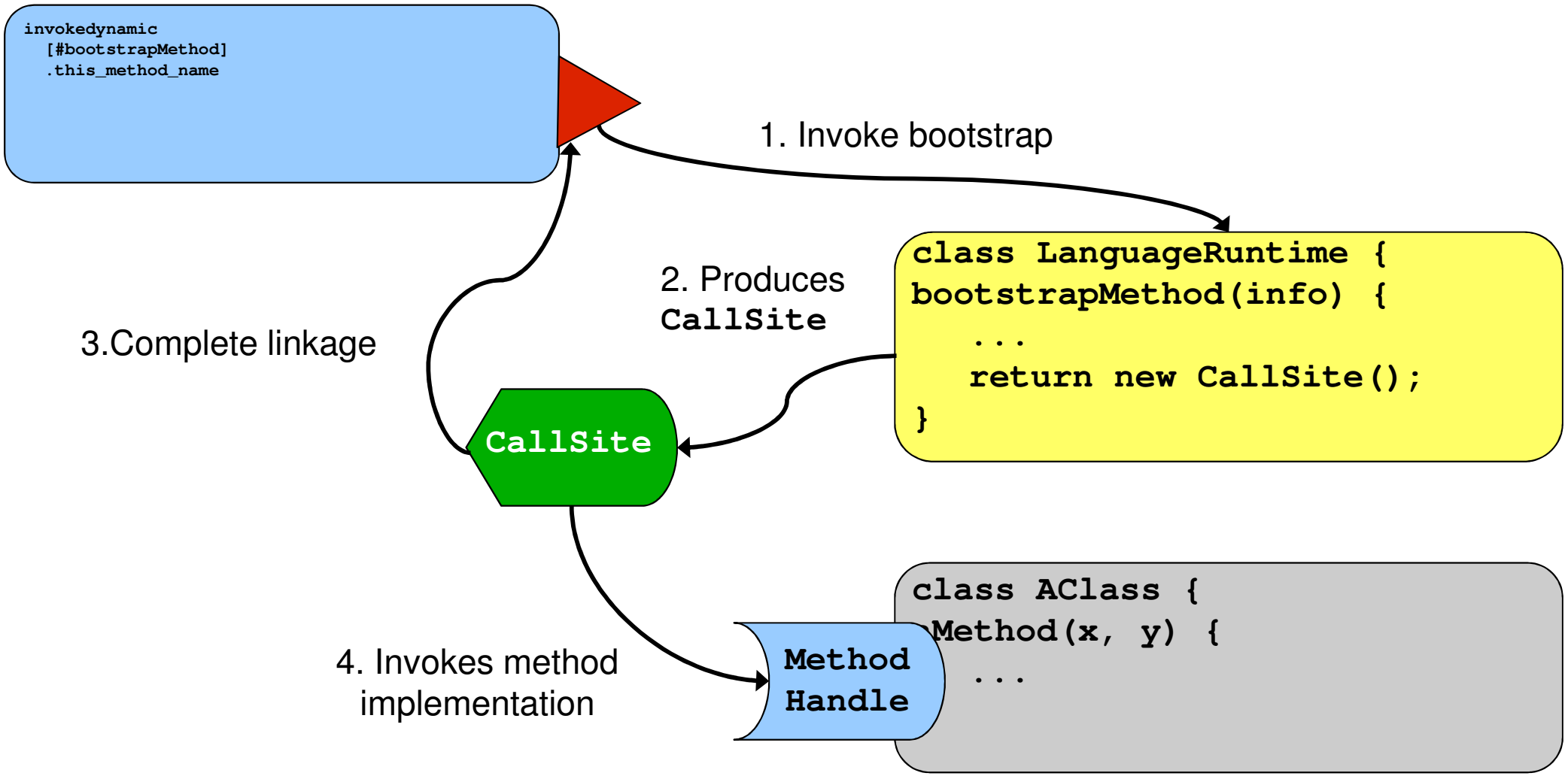
- JVM currently has four ways to invoke method
  - Invokevirtual, invokeinterface, invokestatic, invokespecial
- All require full method signature data
- InvokeDynamic will use method handle
  - Effectively an indirect pointer to the method
- When dynamic method is first called bootstrap code determines method and creates handle
- Subsequent calls simply reference defined handle
- Type changes force a re-compute of the method location and an update to the handle
  - Method call changes are invisible to calling code

`CallSite` and `MethodHandle`

- **`invokedynamic`** linked to a **`CallSite`**
  - **`CallSite`** can be linked or unlinked
  - **`CallSite`** holder of **`MethodHandle`**
- **`MethodHandle`** is a directly executable reference to an underlying method, constructor, field
  - Can transform arguments and return type
  - Transformation – conversion, insertion, deletion, substitution

`invokedynamic` Step 1-to-4

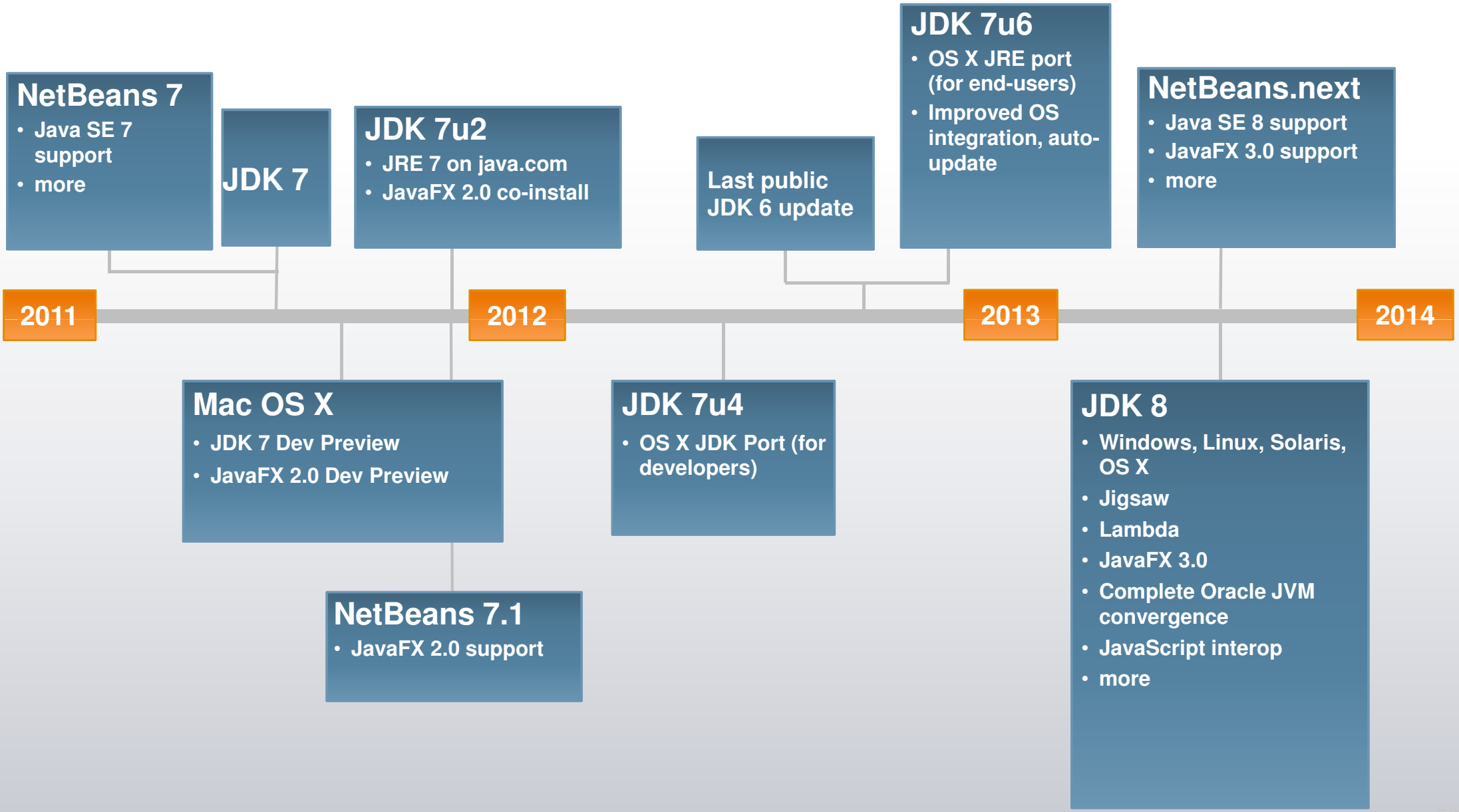
```
this[method_name](x, y)
```



Miscellaneous Things

- Security
  - Elliptic curve cryptography
  - TLS 1.2
- JAXP 1.4.4
- JAX-WS 2.2
- JAXB 2.2
- ClassLoader architecture changes
- `close()` for URLClassLoader
- Javadoc support for CSS

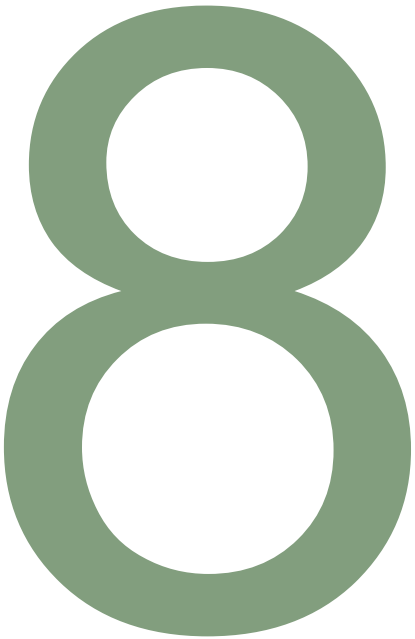
JDK Roadmap


NetBeans 7
• Java SE 7 support
• more
JDK 7
JDK 7u2
• JRE 7 on java.com
• JavaFX 2.0 co-install
Last public JDK 6 update
JDK 7u6
• OS X JRE port (for end-users)
• Improved OS integration, auto-update
NetBeans.next
• Java SE 8 support
• JavaFX 3.0 support
• more
2011
2012
2013
2014
Mac OS X
• JDK 7 Dev Preview
• JavaFX 2.0 Dev Preview
JDK 7u4
• OS X JDK Port (for developers)
JDK 8
• Windows, Linux, Solaris, OS X
• Jigsaw
• Lambda
• JavaFX 3.0
• Complete Oracle JVM convergence
• JavaScript interop
• more
NetBeans 7.1
• JavaFX 2.0 support

JDK 8 geplant für Sommer 2013

- Feedback der Community – **Zwischen den neuen JDK-Versionen werden 2 Jahre Zeit benötigt**
- Release Datum im Sommer 2013 (geändert von 2012)
- Funktionsumfang:
  - Projekt Jigsaw
  - komplette Plattform Modularisierung und Container-Unterstützung
  - Projekt Lambda (JSR 335)
  - JavaScript Interoperabilität
  - Device-Unterstützung
- Weitere kleine Sprachverbesserungen

Projekt Coin Teil 2



ORACLE

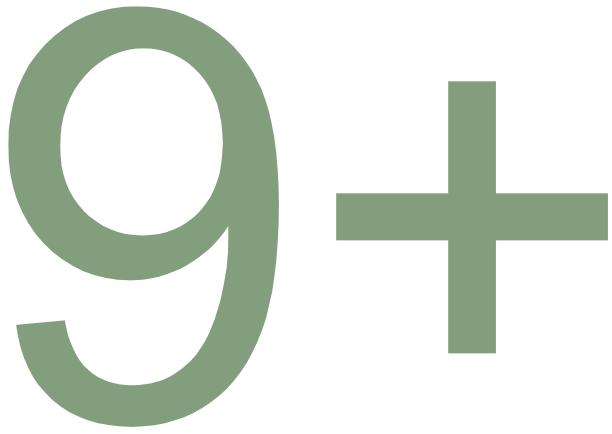
JDK 8 - Sommer 2013

Geplante Inhalte

Thema	Beschreibung/Inhalt
Project Jigsaw	• Module system for Java applications and the Java platform
Project Lambda	• Closures and related features in the Java language (JSR 335) • Bulk parallel operations in Java collections APIs (filter/map/reduce)
Oracle JVM Convergence	• Complete migration of performance and serviceability features from JRockit, including Mission Control and the Flight Recorder
JavaFX 3.0	• Next generation Java client
JavaScript Neu	• Next-gen JavaScript-on-JVM engine (Project Nashorn) • JavaScript/Java interoperability on JVM
Device Support Neu	• Multi-Touch (JavaFX), Camera, Location, Compass and Accelerometer
Developer Productivity	• Annotations on types (JSR 308), Minor language enhancements
API and Other Updates	• Enhancements to Security, Date/Time, (JSR 310) Networking, Internationalization, Accessibility, Packaging/Installation
Open Source	• Open development in OpenJDK, open source additional closed components

JDK 9 und darüber hinaus ..

- Zurück zum Ursprung: Zur tatsächlichen Wirkung der Plattform-Unabhängigkeit von Java
- Verbesserte Interoperabilität mit Non-Java, nativen Sprachen
- Verbesserte Datenintegration von SQL über Name/Value Stores zu Online Feeds
- Verbesserte Device Unterstützung
- Und vieles mehr ... die Arbeiten haben bereits angefangen!



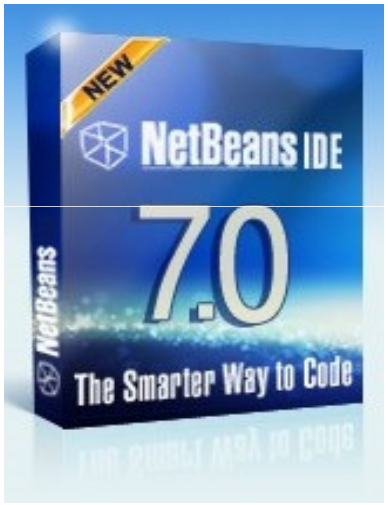
.. die Vision für die Java SE

Interoperability	• Multi-language JVM • Improved Java/Native integration
Cloud	• Multi-tenancy support • Resource management
Ease of Use	• Self-tuning JVM • Language enhancements
Advanced Optimizations	• Unified type system • Data structure optimizations
Works Everywhere and with Everything	• Scale down to embedded, up to massive servers • Support for heterogenuous compute models

ORACLE

NetBeans IDE 7.0.1 und 7.1beta

- Entwicklungswerkzeug für Desktop, mobile und Web Anwendungen
  - **Java EE 6, Java SE 6 (REST, CDI, JPA), Java SE 7, Swing, Java FX**
    - Java Editor support for Project Coin (Diamond Operator, Strings in switch, Multi-catch)
    - Bulk refactoring of projects and packages to Java SE 7
    - JavaFX 2.0 Full edit/compile/debug cycle support
    - Visual debugging of JavaFX apps
  - Java ME und Embedded
  - PHP, Ruby, Groovy, C / C++
  - HTML5 Editing, JSON Formatter
  - Debugger, Profiler, Refactoring



- Läuft auf MS Windows, Linux, Mac OS X und Solaris
- NetBeans IDE ist open-source und frei verfügbar
- Feature Liste & Builds
  - netbeans.org/community/releases/roadmap.html

ORACLE

Unterstützung für eclipce IDE

- Eclipse Projekte mit Werkzeugen und Frameworks
- Ganzheitliche Unterstützung vom Software Development Lebenszyklus
  - Modellierung
  - Entwicklung
  - Deployment Werkzeuge
  - Reporting
  - Daten Manipulation
  - Testing und Profiling
- Primärer Fokus zur Entwicklung von Anwendungen
  - Java EE, Web Services und Web Anwendungen
- Eclipse Unterstützung für andere Programmiersprachen
  - C/C++, PHP, andere

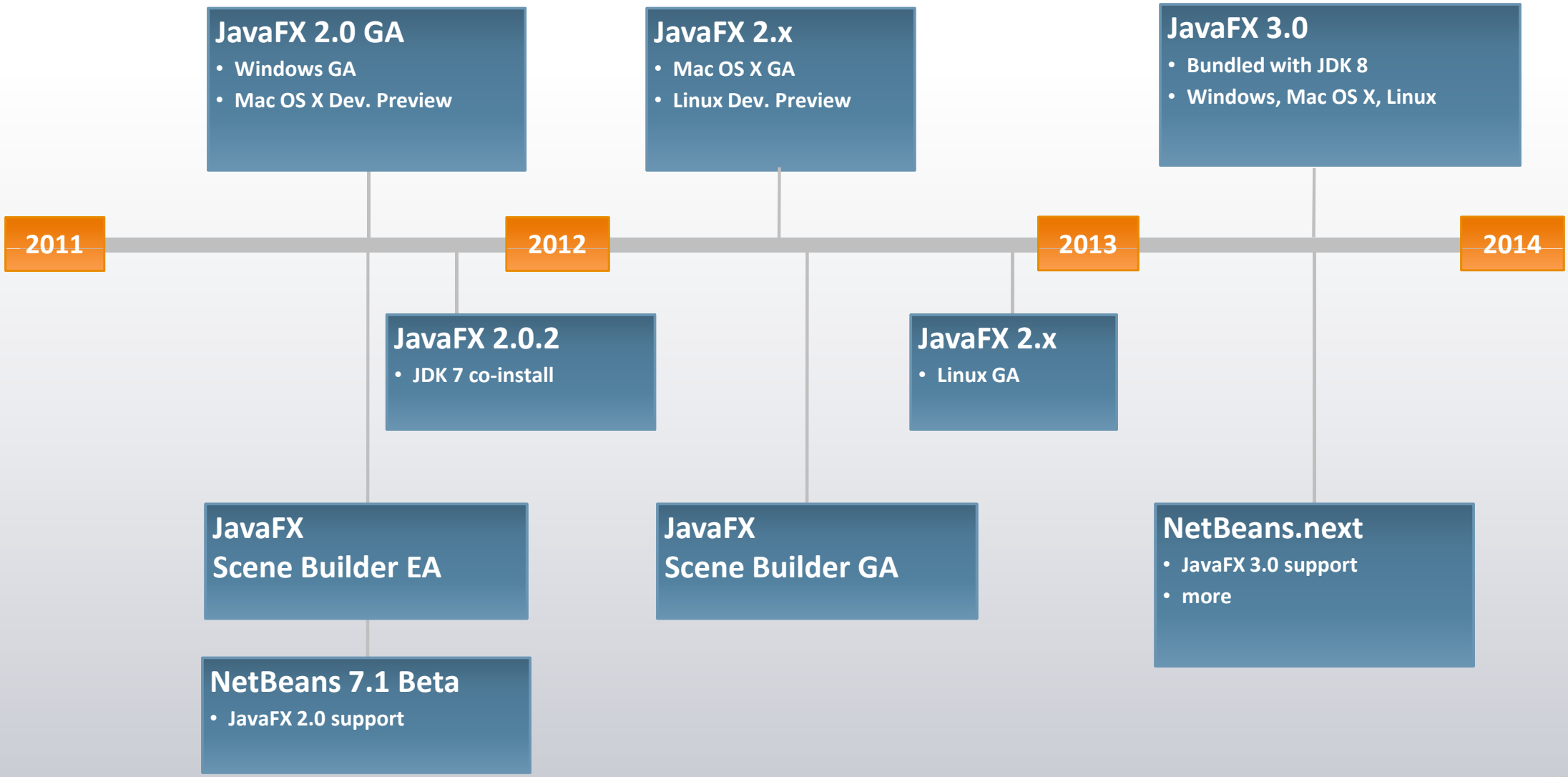
ORACLE

JavaFX 2.0

- **Java/JavaFX als strategische UI für Rich-Client-Applikationen**
- **Einheitliche Applikationsentwicklung für native und Web-Lösungen**
  - Browser Plug-in, Web Start, Native Executables
  - Hardware Accelerated Graphics (DirectX, OpenGL)
  - Nahtloser DOM Zugriff zwischen HTML5 & Java
- **JavaFX 2.0 Plattform Sprachwechsel**
  - Java als native Sprache - anstatt JavaFX Script
  - JavaFX APIs werden in Java implementiert
  - Vorteile bei Verwendung von Generics, Annotations und Multithreading für JavaFX
- **JavaFX 2.0 Release verfügbar**
  - wird bereits vom NetBeans 7.1 Developer Preview unterstützt
  - Migrationspfad für Swing- und SWT-basierte Anwendungen
  - wird Open Source und im JCP standardisiert

ORACLE

JavaFX Roadmap



Design Ziele für Java ME

Abstand von Java SE zu Java ME verringern

- Synchronize CLDC and JDK releases
- Converge CDC and Java SE Embedded

Volle Einbeziehung vom Embedded Markt

- Java Platform covering all CPU/Footprint variants
- Dedicated APIs for vertical market segments

Tiefgreifende Integration von Content und Services

- Services embedded into Oracle runtimes and tools
- Developer/consumer access to carrier services

Java ME und Java SE rücken näher zusammen

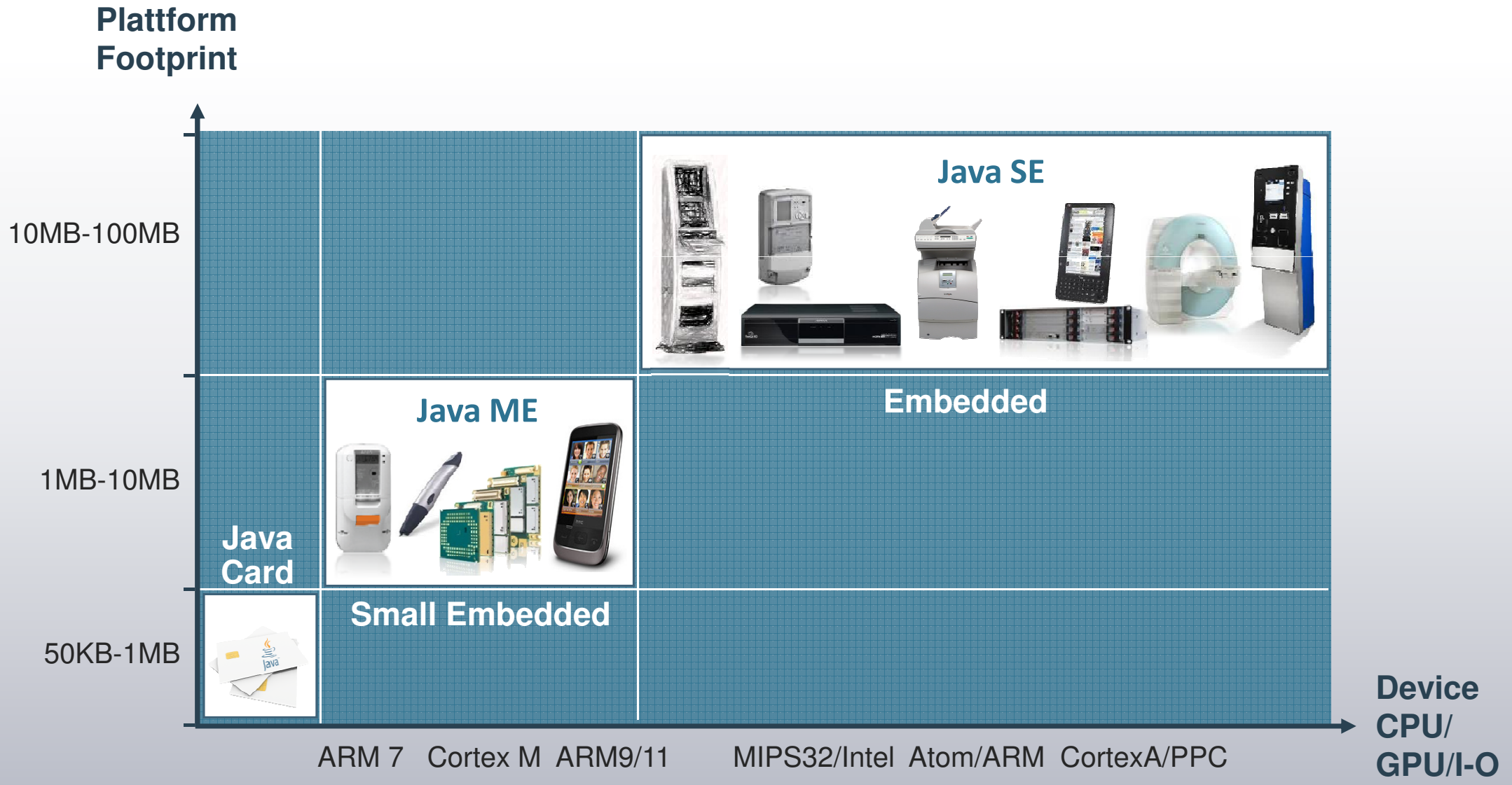
Java ME 7 und Java ME 8

- Java ME alignment with SE
  - Synchronized releases
  - Latest Java language support
  - Java ME APIs can run on Java SE
  - Consistent tool interfaces
- New APIs for mobile phones and billions of connected devices

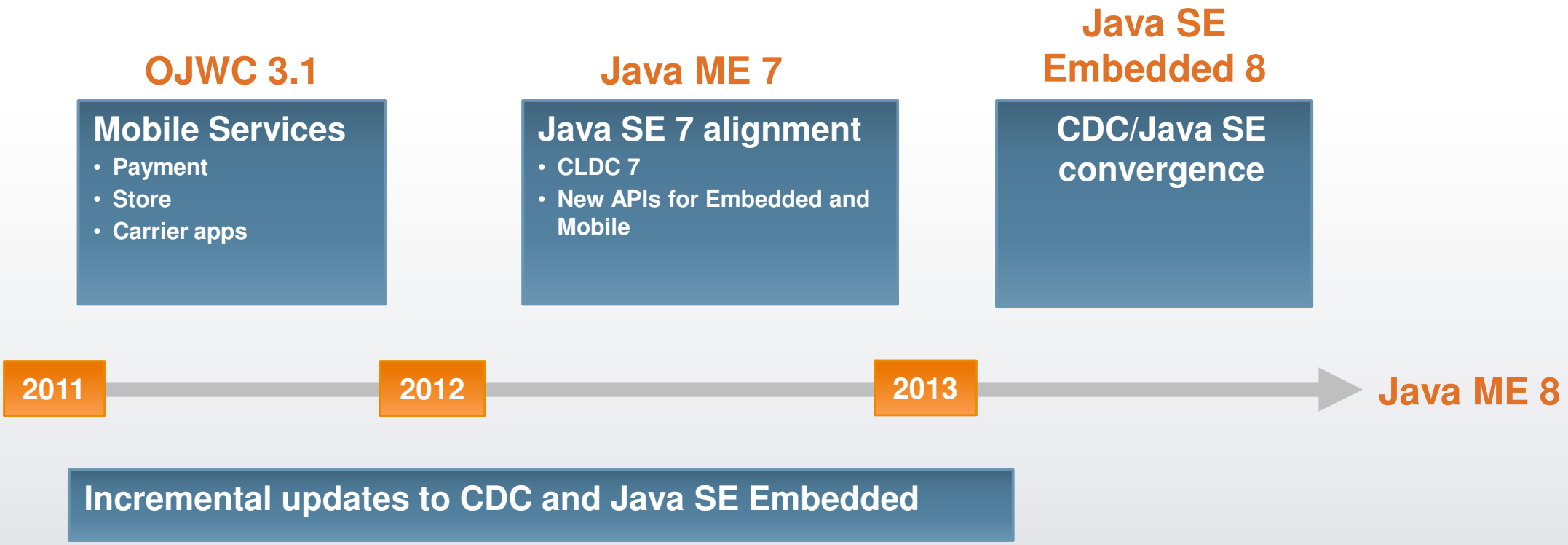
CDC/Java SE 8 Convergence

- "CDC Profile" in SE 8
  - Porting CVM features to Hotspot JVM
  - JDK 8 libraries with smaller and faster variants/subsets
- JavaFX as graphics framework
- Best features and performance from CDC and SE for Embedded

Java Technologie für Embedded Device - 2013



Java ME/Embedded Roadmap



Developer Tools zum Download verfügbar:

- **Java ME SDK 3.0.5**
- **LWUIT 1.5**

Design Ziele für künftige Java EE

Standard zur Entwicklung von Enterprise Anwendungen

Von kleineren Web Anwendungen bis zu hochskalierbaren Multi-tier Enterprise Anwendungen

Verbesserte Produktivität für Enterprise Java Entwickler

Vereinfachtes und umfassendes Programmiermodell

Portabilität: Hersteller- und Infrastruktur-übergreifend

Anpassungsfähig: Einbinden von Non-Java EE Frameworks

Granularität: Unterstützung für modulares Design

ORACLE

Wie sich die Entwicklung durch die Wolke schrittweise verändert

- **Entwickler für Unternehmensanwendungen wollen Cloud-Lösungen von der eigenen IT-Abteilung**
  - IaaS als neuer “Self-Service Data Center”
  - Unmittelbar, On-Demand Provisioning
  - Hosted, sichere Cloud Services

- **Virtualisierung ist ein wertvoller technischer “Building Block”, aber keine Plattform**
- **Entwickler schauen nach einem PaaS Standard für die nächste Generation Cloud-basierter Anwendungen**
  - Die Java EE Plattform eignet sich als PaaS Standard
- **Java EE hat bereits vergleichbare Herausforderungen für die IT gelöst**

Java EE 7 und Java EE 8 – Themenschwerpunkte

Cloud	Modularität *	HTML5
• Provisioning	• Building on Jigsaw	• Emerging Web Standards require a programming model
• Elastic & Autonomic Scalability	• Focus on OSGi interop	• JSON, WebSockets, off-line, APIs & DOM
• Multi-Tenancy	• Supporting Profiles & Modular Applications	

* Aligning with delivery schedules of Jigsaw in Java SE 8

Java EE 7 – Provisioning

Java EE Vorhandenes Modell

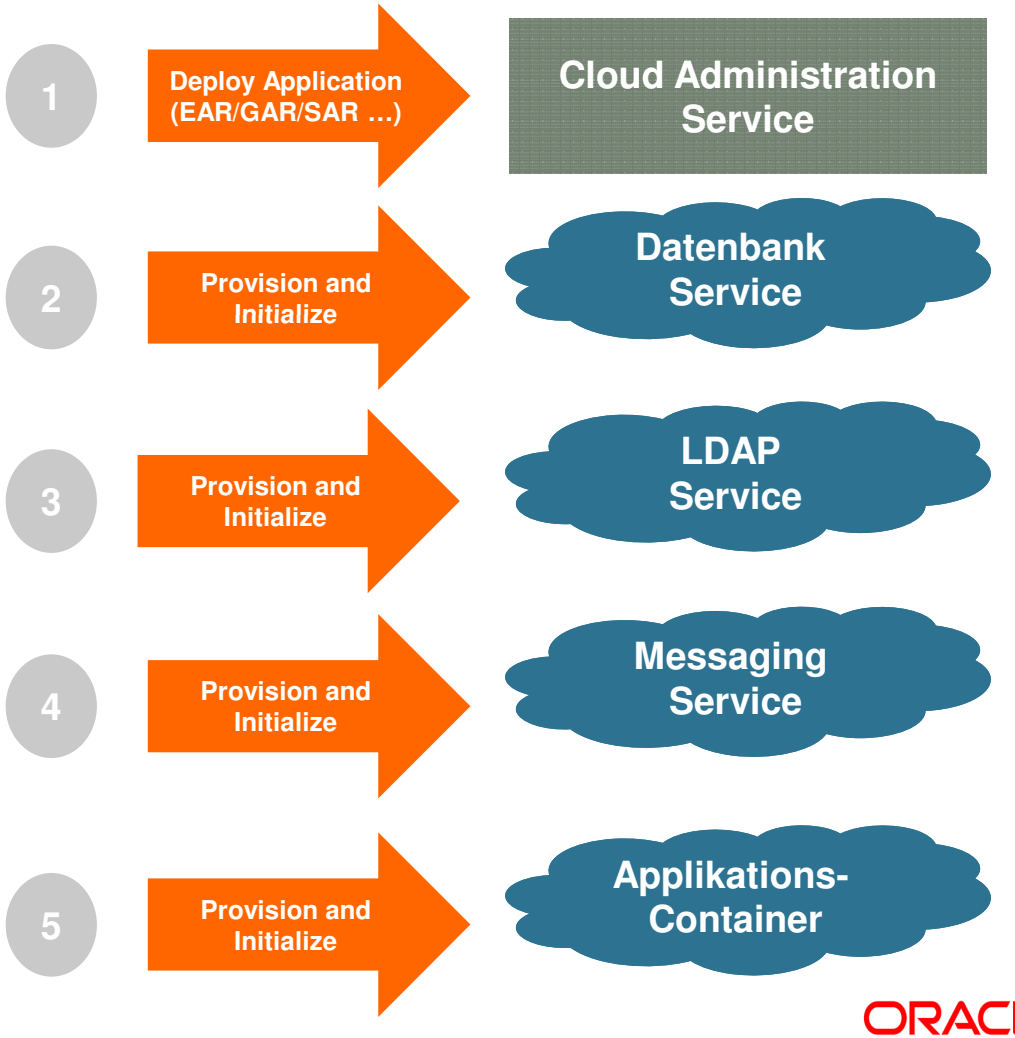
- Konfiguration Java EE Ressourcen – JDBC, JMS, etc.
- Deploy Application Archive (.EAR)

Java EE 7 Modell

- Auto-Provision Services von Application Dependencies
  e.g. Datenbank, LDAP

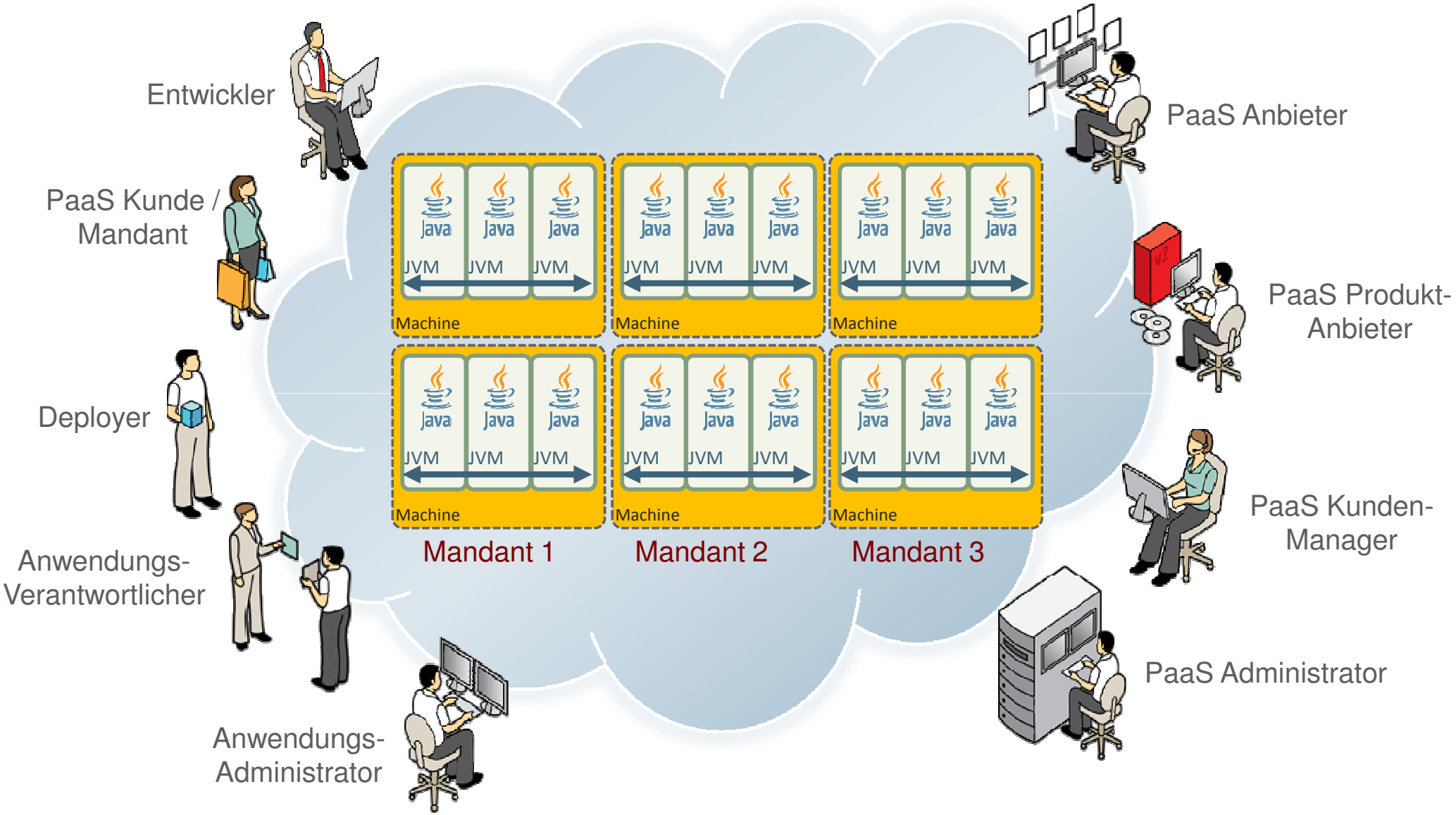
Extensible Deployment Modelle für Frameworks

- Spring, Seam, etc.



ORACLE

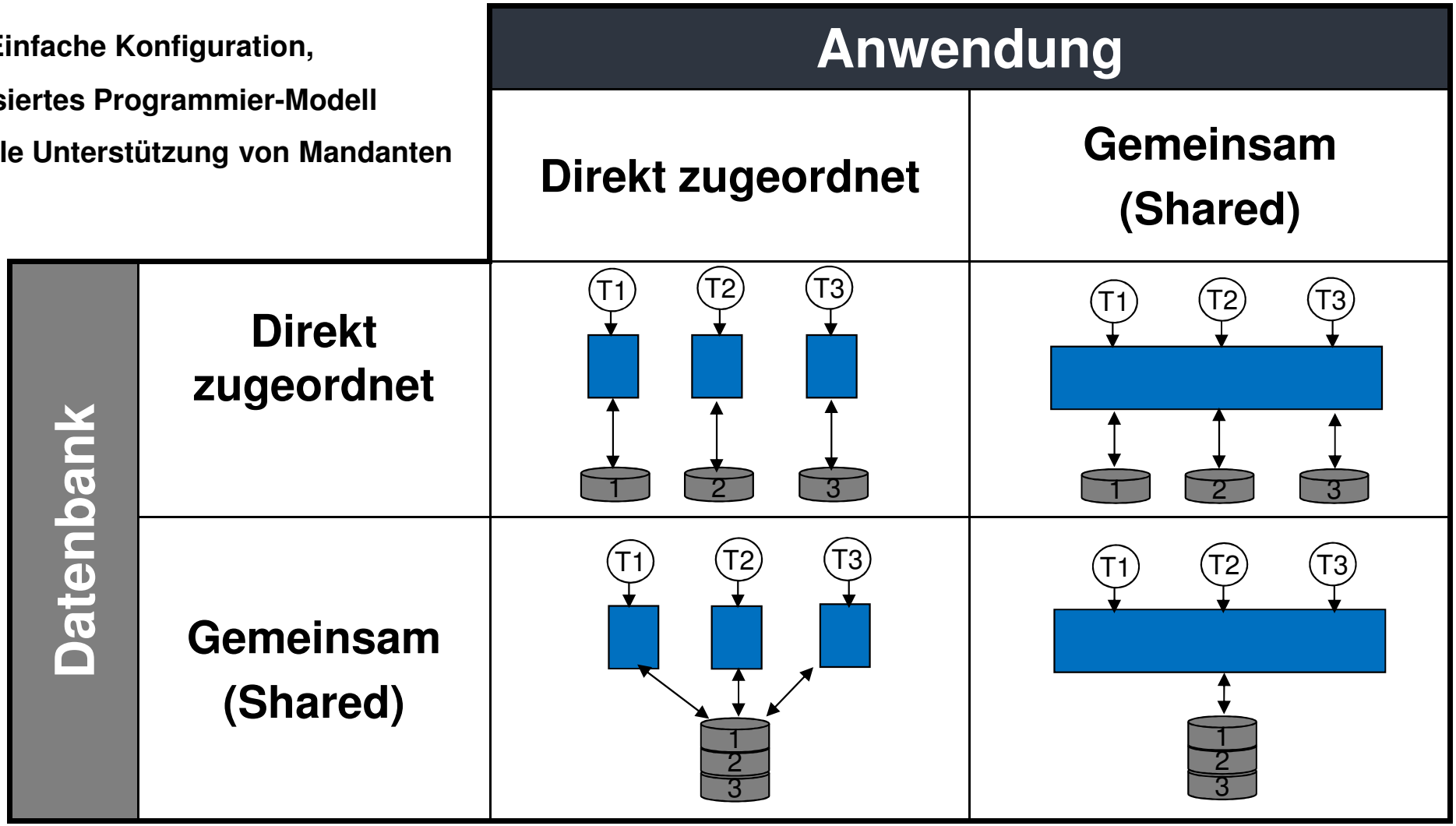
Java EE 7 - Rollenmodell



ORACLE

Java EE 7: Persistenz und Mandantenfähigkeit

- **Ziel: Einfache Konfiguration, CDI-basiertes Programmier-Modell**
- **Flexible Unterstützung von Mandanten**



Einzelanwendungs-Deployment mit Unterstützung für unterschiedliche Mandanten-Architekturen (Multi-Tenancy)

Java EE 7 – Geplante Inhalte

Thema	Beschreibung/Inhalt
PaaS Enablement	• Service Definitions and Provisioning to enable Java as Platform as a Service • Enable Multi-tenancy in APIs
Web Profile	• Provide popular additions to the Web Profile including JAX RS 2.0 Support
JMS 2.0	• Simplify the programing model for building messaging based applications • Dependency Injection support
CDI	• Tighter Integration with JSF • Expand scope of container managed transactions • Expanded service metadata and improved configuration
Caching	• Provide APIs for accessing caching systems
Concurrency Utilities	• Support for Java concurrency APIs within the container
Pruning	• Allow vendors to optionaly support older APIs • EJB CMP/BMP, JAX-RPC
Open Source and Transparency	• Open development under project GlassFish on java.net • Java EE 7 JSRs run in open with publicly viewable Expert Group mail archive

Java EE 7 JSRs

- **Plattform 7 / Web Profile 7**
- **JPA 2.1**
- **JAX-RS 2.0**
- **EJB 3.2**
- **JMS 2.0**
- **Servlet 3.1**
- **EL 3.0**
- **JSF 2.2**
- **CDI 1.1**
- **Bean Validation 1.1**
- **JCache 1.0 (JSR 107)**
- **Concurrency Utilities 1.0**
- **State Management 1.0**
- **Batch Processing 1.0**
- **JSON 1.0**
- **Common Annotations 1.2 MR**
- **JAX-WS 2.3 MR**
- **JTA 1.2 MR**
- **JSP 2.3 MR**
- **JASPIC 1.2 MR**

➢ Die Arbeiten an den Java EE 7 JSRs sind öffentlich: siehe java.net
  – http://javaee-spec.java.net
  – Eigenes Projekt pro Spezifikation: jpa-spec, jax-rs-spec, jms-spec, …

Zusammenfassung

- Die Java Plattform nutzt Innovationen im Ökosystem und wird sich weiterentwickeln
- Java Plattform liefert eine echten Mehrwert
- Unternehmen profitieren von existierende Investitionen in Java EE
- Java SE 8 wird einige neue und größere Features beinhalten
- Die Ausrichtung für Java SE 9 läuft bereits

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Vielen Dank für Ihre Aufmerksamkeit!

Wolfgang.Weigend@oracle.com