

# The Java IP Story

Prof. Dr. Dirk Riehle, M.B.A.

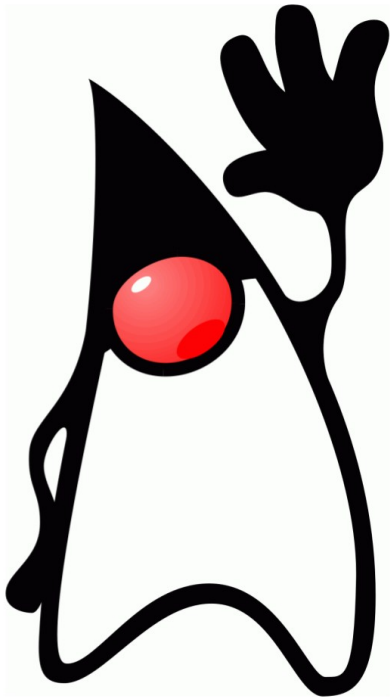
Friedrich-Alexander University of Erlangen-Nürnberg

**FAU – AMOS – SS 2011**

Java is an important technology powering the modern web and in particular enterprise applications. It has a checkered intellectual property history, and with the recent acquisition of Sun, the Java creator and owner, by Oracle, things only stand to heat up. This slide set discusses some of the more interesting issues around Java intellectual property and its strategic use in business.

1. What is Java?
2. Short Java IP Story Time-Line
3. Three Substories
4. Java's Challenge to the Windows Platform
5. Microsoft and Java
6. The OpenJDK Strategy (Open Core Model)
7. Certification of Compatible Implementations
8. Threats to Commercial Revenue
9. Main Tools to Curtail “Competitors”
10. Problems for Alternative Implementations
11. Problems for OpenJDK Forks
12. Thank you! and References

# What is Java?



**Java**  
**SDK / JRE**  
**JCP / JSPA / JSR**

2

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

"Java" consists of multiple components [2].

- **Java**, the programming language, which has been standardized
- the **JRE**, which contains the necessary infrastructure to run Java applications
- the **JDK**, which is the JRE including its libraries and runtime, plus some tools
- in addition, there are many third party libraries and development tools.

Java comes in multiple editions, which target different execution environments, most notably embedded, desktop, and enterprise systems.

Java, the programming language, is changing only slowly. However, the libraries are evolving at a rapid pace. To make them useful to industry, the **JCP**, the Java Community Process, defines specifications and develops reference implementations and compatibility test suites. The JCP was set up by Sun and brought together all relevant industry players with an interest in Java. To be allowed into the process, vendors have to sign the **JSPA**, the Java Specification Participation Agreement. The individual specifications are called the **JSRs** for Java Specification Requests.

Originally Sun, now Oracle, owns the following intellectual property rights:

- the Java trademark; this allows it to stop labeling other parties software as "Java" in the domain of information technology
- various JSR test suites; this output of the JCP serves to certify Java component implementations as specification compatible
- many patents in the implementation of the core Java programming language and runtime

These property rights are being used in various scenarios.

## Short Java IP Story Time-line

- 1996** Java released under proprietary license
- 1998** JCP announced, formalized
- 1998** First alternative Java implementation (GNU Classpath)
- 2001** Sun wins 1997-filed trademark lawsuit against Microsoft
- 2005** Apache Harmony project started
- 2007** Sun finishes releasing Java under GPLv2, OpenJDK
- 2007** Java TCK includes field-of-use restrictions
- 2010** Oracle acquires Sun
- 2010** IBM leaves Apache Harmony for OpenJDK
- 2010** Apache Software Foundation protests JCP

3

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

To get started, let's review a few key events in Java's history [1]. Java was released to the public in 1996 under a proprietary license. Developers could use Java, but it wasn't open source. With fast-rising industry interest, Sun announced and formalized the JCP in 1998 to help move Java forward by involving other industry players.

Microsoft was one of the industry players that adopted Java on its platform early on. Already in 1997 Sun filed a lawsuit alleging that Microsoft was not playing by the licensing agreement, and in 2001 Sun won this lawsuit. Microsoft has since removed Java from its platform leaving the integration to 3rd parties.

A first open source implementation of the Java tools, runtime, and libraries was begun in 1998 as the GNU Classpath project; it has yet to reach completion. The project uses the GPL open source license with a modification, the classpath exception, that allows linking of applications so that they don't fall under the GPL.

Another open source implementation of Java was begun under the auspices of the **ASF**, the Apache Software Foundation, in 2005, labeled "Project Harmony". Sun, now Oracle, offers the non-profit community free Java certification if it was willing to accept certain field-of-use restriction for its software. The ASF objected on principled (open source) grounds, preventing it from using the Java trademark.

Sun itself released most of the Java tools, libraries, and runtime in 2006, finishing it in 2007, under the GPLv2 open source license. Called the OpenJDK, this is still the only complete open source implementation of Java available today.

# Microsoft and Java

## Field-of-use Restrictions

## OpenJDK and Patents

4

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

I am going to tell three stories, in which intellectual property and business strategy come together:

- The Java vs Windows platform war
- Generating Revenues with the Open Core Model
- Curtailing Competition using Patents and Field-of-use restrictions



Questions of intellectual property and the legal system have a habit of boring one's brain out; I hope this won't happen in this case.

# Java's Challenge to the Windows Platform

**“Write once, run anywhere.”**

6

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

Java's original marketing slogan (and promise) to software developers was to "write once, run anywhere" (WORA). It points to the most basic of industry strategies, the platform strategy.

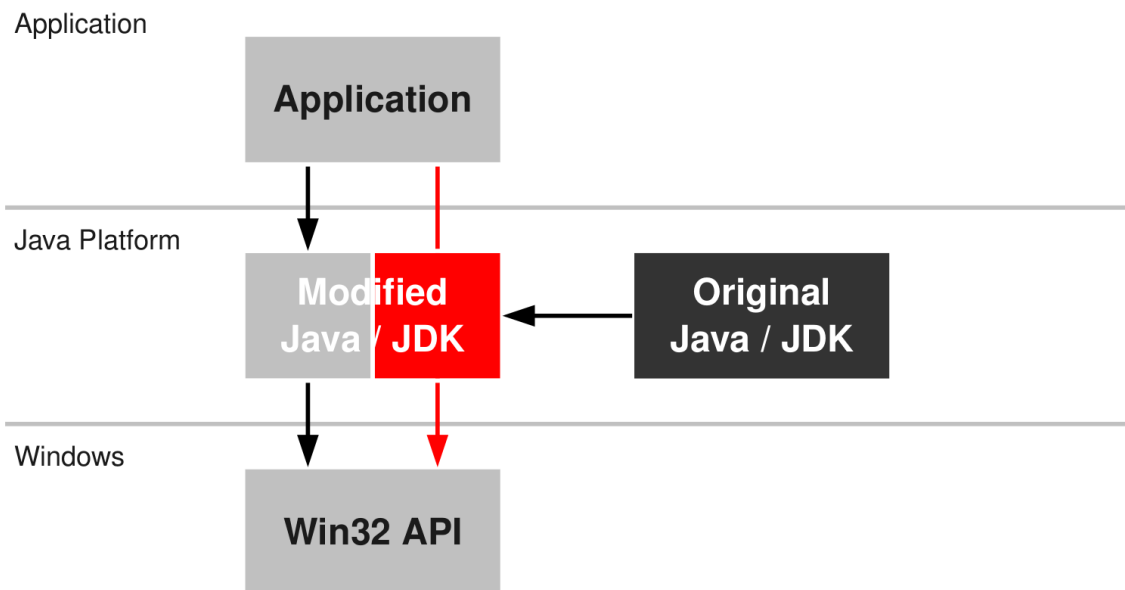
In a platform strategy, a software vendor tries to make its software become the platform that everyone else builds on. By building applications on top of someone's platform, these applications become dependent on the platform, and whenever a copy of the application is bought, a copy of the platform has to be bought too.

This is why Windows is so powerful and Bill Gates became so rich.

Java now challenged the Windows platform by completely hiding the Windows programming interface, the Win32 API, behind the Java class libraries. Thus, applications could be implemented using Java class libraries only without knowing whether these are run on a Windows or a Linux platform. This made applications portable across different operating systems and led to Java's rallying cry of "write once, run anywhere".

Of course, Java was simply trying to become the next platform to build on by hiding the underlying operating system, be it Windows or MacOS. Scott McNealy, the Sun CEO, probably wanted to become as rich as Bill Gates.

# Microsoft and Java



7

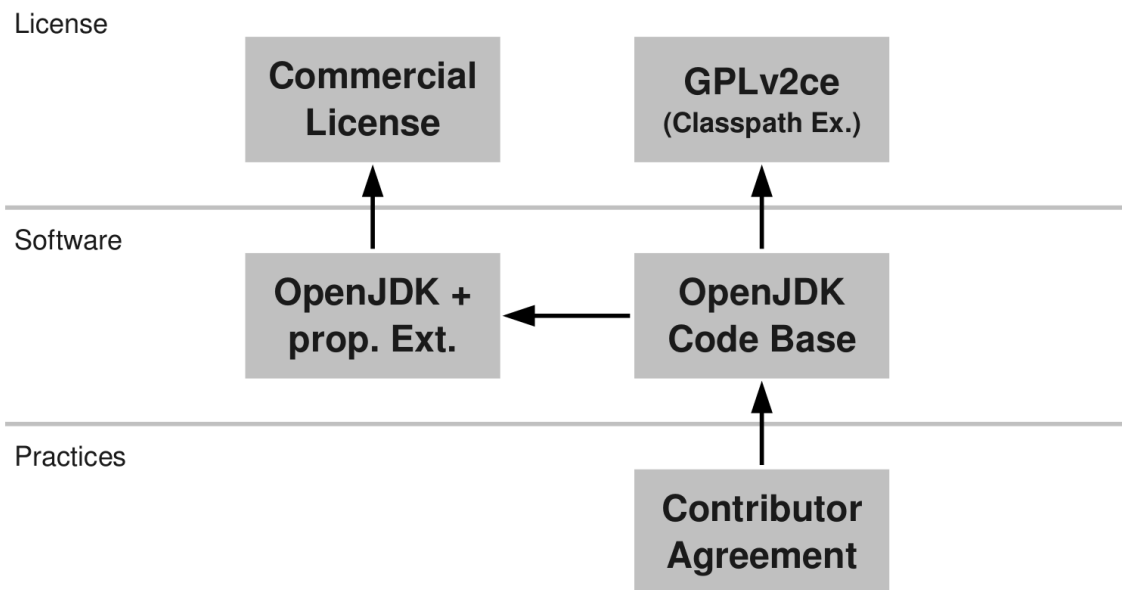
FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

Microsoft didn't like Sun's Java strategy as it would reduce the lock-in of applications to the Win32 API and hence make application developers less dependent on Windows. Still, customers kept asking for Java on Windows. Eventually, Microsoft complied: Microsoft licensed Java from Sun and deployed it on the Windows platform.

According to a 1997 lawsuit by Sun, Microsoft did so only by violating the license agreement. Sun stated that Microsoft modified the JDK by omitting libraries that were needed to run Java well. This required application developers to go directly to the Win32 API. This strategic move was to maintain the lock-in to the underlying Windows platform that Java was trying to do away with in the first place.

Sun won the lawsuit in 2001, requiring Microsoft to pay damages. In the aftermath, Microsoft removed Java from its platform and introduced alternative technologies. Today, Java on Windows is provided by third parties only [3].

# The OpenJDK Strategy (Dual-License Model)



8

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

Knowing that Sun released Java as open source in 2007 and knowing that there were alternative implementations, you may be wondering how Sun ever intended to make money on Java. One answer to this is called the [open core model](#) today.

The release of the OpenJDK by Sun in 2007 gives developers a full Java development and runtime environment under the GPLv2 open source license. The license even comes with a modification called the Classpath Exception that allows software vendors to combine their application with the OpenJDK without having to open source their application code.

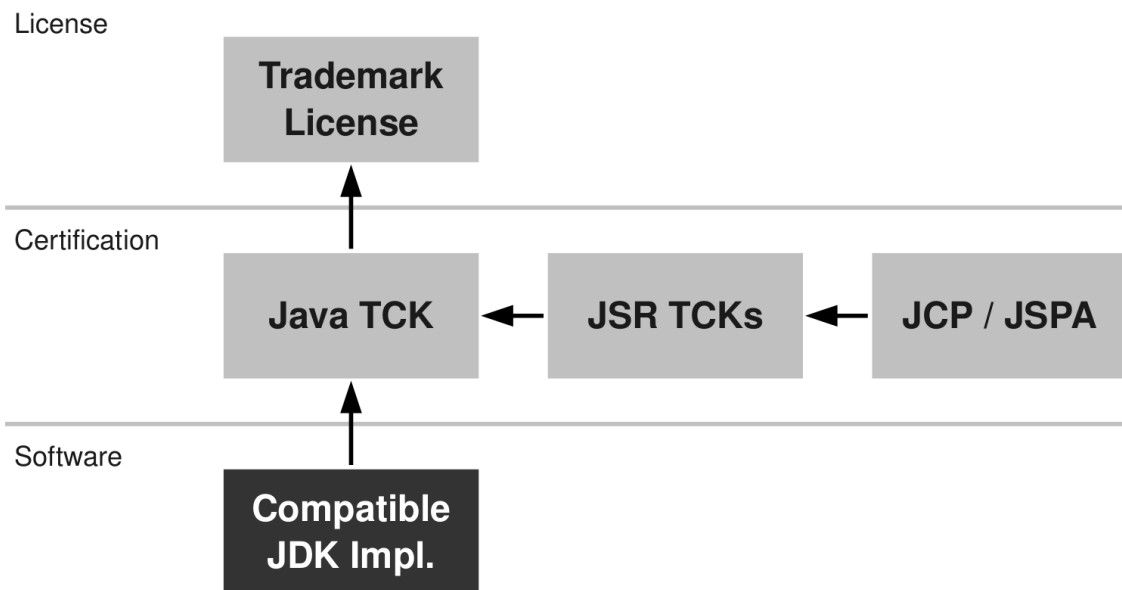
To not fully lose their (potential) revenue stream, Sun does two things, which taken together represent the core intellectual property strategy behind the open core model:

- Sun maintains the full copyright and thereby ownership of the OpenJDK code base. Any outside contributors first have to sign a joint copyright agreement [8]. While this curtails code contributions, it ensures that Sun remains the owner of the JDK.
- Sun then uses their ownership rights to offer the OpenJDK under a commercial license to those who don't like the GPL (dual license strategy) while at the same time providing additional non-open-sourced functionality (open core model).

Thus, to get full service as well as enhanced features, Java application developers have to pay for the commercial license rather than use the open source license.



# Certification of Compatible Implementations



9

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

To win against Microsoft early on, Java needed to achieve rapid growth and industry buy-in during the 1990s.

Java could only grow as fast as it did, because Sun brought other major industry players on board. This was done through the JCP, the Java Community Process. In that process, companies like IBM, SAP, and anyone else who cares can influence the development of Java through library specifications and make sure it works well with their products.

One tangible output of the JCP is a stream of JSRs, the Java Specification Requests. Each JSR specifies a library or similar component. To each JSR specification belongs a TCK, the technology compatibility kit, for the JSR. A TCK is effectively a test suite that checks a given implementation of the library for compliance with the specification. If the library passes, it is considered a compliant implementation. A Java TCK combines various JSR TCKs into a full test suite for a particular Java configuration.

Anyone with a Java implementation who wants a Java trademark license from Oracle first has to pass the matching Java TCK.

## Enhanced OpenJDK Forks

## Alternative Implementations

10

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

Rapid growth and the open core model meant that there could be serious business.

There was only one threat to those revenues: Alternative implementations. These are implementations of Java, including a JVM, tools, and class libraries. The first of these implementations, the GNU Classpath project, started as early as 1998 [17]. A later alternative implementation is the Apache Harmony project [16].

Because of the broad industry involvement, Sun and later Oracle couldn't simply close off Java again after it grew to dominance as the leading enterprise programming language. Thus, Oracle today provides a defined way for industry to acquire a Java trademark license if it wants to label their Java implementation “Java”.

Today, Oracle makes the Java trademark license available in two main variants:

1. For paying customers, a full trademark license is granted, if the Java implementation passes the relevant Java TCK
2. For non-paying open source projects like GNU Classpath or Apache Harmony, a restricted license is granted

Option 1 works very well for Oracle, after all, it receives presumably appropriate revenues. Option 2 is not really an option for open source projects as we will see, but rather an attempt to prevent these projects from gaining traction.

# Lack of Patent Provisions in GPLv2

## Field-of-use Restrictions in Java TCK

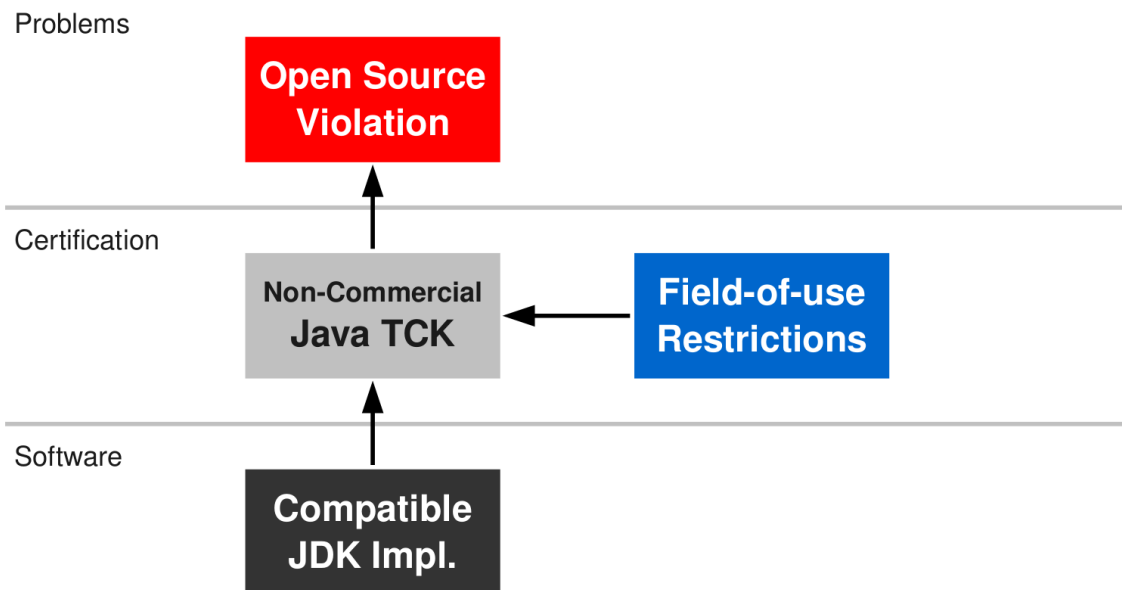
11

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

With the OpenJDK, Oracle provides a full open source implementation of Java. However, in addition, Sun, now Oracle, made two strategic moves to curtail alternative implementations:

- The choice of the (1991) GPLv2 license because of its poor coverage of patents
- The field-of-use restrictions in the trademark license for open source projects

# Problems for Alternative Implementations



12

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

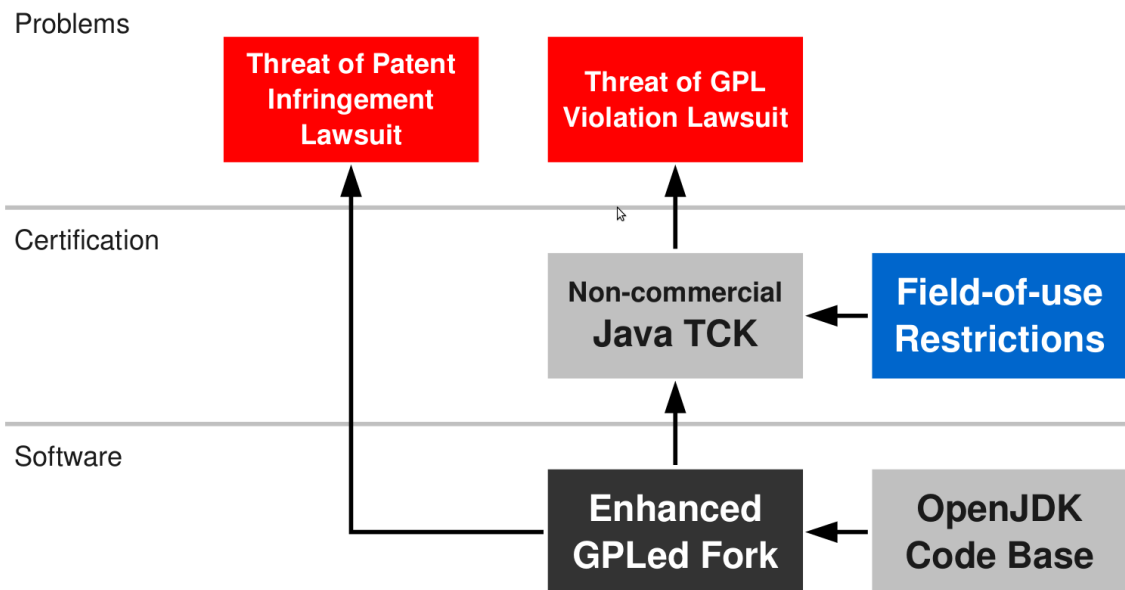
An open source Java implementation that wants to call itself “Java” has to pass the appropriate Java TCK. The use of the TCK and the Java trademark license is only granted for free, if the project (more precisely the organization behind it) accepts certain field-of-use restrictions for the project. As of today (2011), the main restriction is that the alternative implementation may not be used for embedded systems development. Enterprise applications, in contrast, are fine.

An example of an alternative implementation is the Apache Harmony project, which provides a comprehensive Java implementation under the Apache License 2.0.

The definition of Open Source Software requires that no field-of-use restrictions be imposed on the user. Oracle's field-of-use restrictions are not acceptable to an open source project that wants to call itself that way [11] [12]. (And the same applies to free software.)

Game over for unrestricted free and open source Java implementations.

# Problems for OpenJDK Forks



13

FAU - AMOS - SS 2011  
© 2011 Dirk Riehle - Some Rights Reserved

But what about forking the OpenJDK, wresting it off the hands of Oracle?

In forking the OpenJDK, you face two options:

- If you want to put the “Java” label on your fork, you'll have to pass the TCK and accept Oracle's field-of-use restrictions. Since you are required to keep the software under the original GPLv2 license, this may hand you a GPL violation lawsuit, after all, you just accepted field-of-use restrictions which are incompatible with the license [18].
- If you don't care about the “Java” label, you can forgo certification. However, without the certification, you don't get a license to the software patents buried in the OpenJDK code. Thus, any patent owner who happens not to like you might decide to sue your customers who will turn back on you and will find you without patent defense.

One hope is Google's Dalvik, once it digs itself out of the siege of patent lawsuits it is being buried under, but that will then be the Dalvik IP Story, not the Java IP story...

Game over, again.

For now.

# Thank you! Questions?

dirk.riehle@cs.fau.de - <http://osr.cs.fau.de>

dirk@riehle.org - <http://dirkriehle.com> - @dirkriehle

- [1] Oracle. [The Java History Timeline](#).
- [2] Wikipedia. [Java \(Programming Language\)](#).
- [3] Microsoft. [Microsoft Java Virtual Machine Support](#).
- [4] Oracle. [GNU General Public License, version 2, with the Classpath Exception](#).
- [5] Wikipedia. [GPL Linking Exception](#).
- [6] Oracle. [Oracle and IBM Collaborate to Accelerate Java Innovation Through OpenJDK](#).
- [7] Oracle. [Java SE at a Glance](#).
- [8] Oracle. [How to Contribute](#).
- [9] Oracle. [OpenJDK FAQ](#).
- [10] Doug Lea. [Email to JCP Executive Committee Members](#).
- [11] Apache Software Foundation. [Our Participation in the Java Community Process](#).
- [12] Apache Software Foundation. [Open Letter to Sun Microsystems](#).
- [13] Paul Querna. [Java Trap, 2010 Edition](#).
- [14] Oracle. [Java\(tm\) Compatibility Test Tools \(Java CTT\) Info](#).
- [15] Wikipedia. [Technology Compatibility Kit](#).
- [16] Wikipedia. [Apache Harmony](#).
- [17] Wikipedia. [GNU Classpath](#).
- [18] Brian McCallister. [The TCK Trap](#).
- [19] Stefano Mazzocchi. [Dalvik: How Google Routed Around Sun's IP-based Licensing Restrictions on Java ME](#).