

# Adaptable Runtime Monitoring for the Java Virtual Machine

Nowadays, many programming language implementations and programming frameworks target the Java Virtual Machine (JVM). Examples include the Java and Scala compilers, Oracle's Truffle framework and the interpreters built on top of it for a variety of dynamic programming languages, as well as big-data frameworks such as Apache Spark, Apache Flink, and Apache Storm. Unfortunately, the JVM provides only limited support for runtime monitoring. The JVM Tool Interface (JVMTI) offers only a very low-level programming model, often introduces high overhead, and does not guarantee proper isolation of the monitoring logic from the observed program. Aspect-Oriented Programming (AOP), in particular AspectJ, is often used to implement runtime monitoring tools. While offering a convenient programming model, the use of such technologies exacerbates performance- and isolation-related problems. In this paper, we advocate the use of our dynamic program analysis framework DiSL for runtime monitoring on the JVM. DiSL reconciles an AOP-based programming model, full coverage of all executed bytecodes, optimizations of the monitoring logic, and support for isolation of the monitoring logic. Moreover, DiSL also offers an API to deploy, adapt, and remove monitoring logic at runtime, and it provides seamless support for monitoring also applications running on Android devices.

Author: Walter Binder ([walter.binder@usi.ch](mailto:walter.binder@usi.ch))