

Secure, Pervasive Sharing of Language Runtimes in Cloud Applications

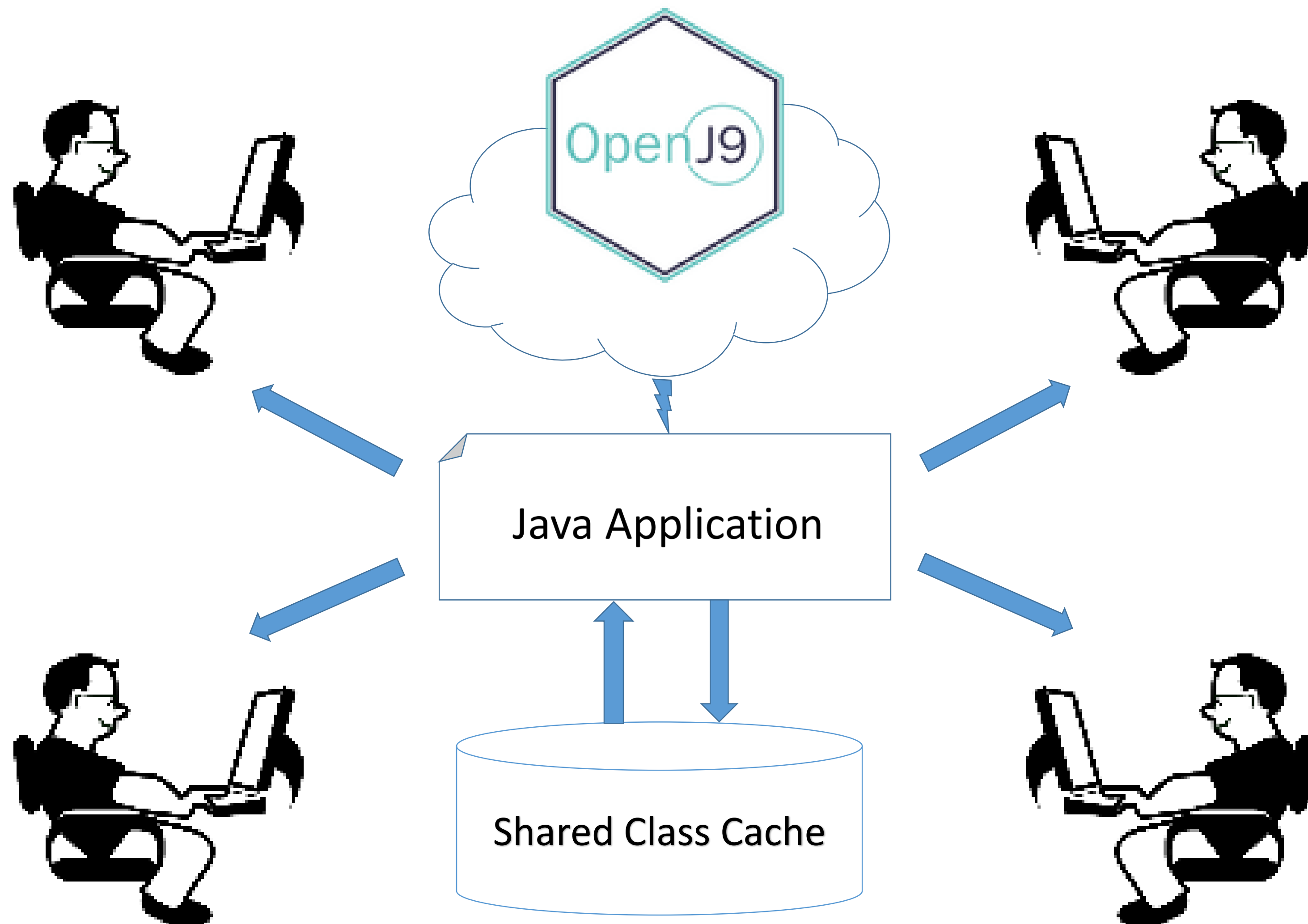
Mark Thom, Gerhard W. Dueck, Kenneth B. Kent
University of New Brunswick, Faculty of Computer Science

Daryl Maier

IBM Canada

{mark.thom, gdueck, ken}@unb.ca, daryl.maier@ibm.com

Cloud Computing with Eclipse OpenJ9



The Shared Cache delivers the following benefits to Java applications running on Eclipse OpenJ9, among others:

- Immediate dissemination of application updates to cloud clients
- Hastening of application startup times by storing pre-compiled bootstrap code
- Drastically reduced memory consumption by caching common data and code to the cloud
- Robust support for these and other features across many prominent operating systems and machine architectures

But what about applications written in languages other than Java? Could we use the same technology to deliver these advantages to emerging language runtimes?

The Eclipse OMR Project (<http://eclipse.org/omr>)

- Spawned from the Eclipse OpenJ9 Java Virtual Machine
- Purpose: provide robust, reusable components for deployment in future compilers and language runtimes
- Contains many components at varying stages of completion:
 - Runtime diagnostic tools
 - Garbage collectors
 - Just-in-time (JIT) compiler
 - Support for embedded environments and cloud computing??



Motivation: a language-agnostic shared cache

Eclipse OMR has successfully imported its powerful JIT compiler technology from OpenJ9. However, there are resource-constrained settings – such as embedded systems – where JIT compilation is prohibitively expensive. Traditionally, this has inhibited embedded systems development in productive, high-level languages. The shared cache offers a convenient workaround to this: application code is pre-compiled, stored to the cache, and made available for loading through the cloud.

We are adapting OpenJ9's shared cache technology for use in other language runtimes. It is divided into the following functional layers.

Language Runtime Interface

Query system for retrieval from cache
Invalidation of obsolete data and code

Storage Policy Layer

Allocation and layout policies
Concurrency and memory protection features

Operating System Layer

Persistent/non-persistent semantics
User access & security policies
Cross-platform compatibility library

