Performance and Security Improvements

Sander Mak
FELLOW & SOFTWARE ARCHITECT
@Sander_Mak

Garbage Collector Deprecations

Removed: GC combinations deprecated in Java 8

Deprecated: Concurrent Mark Sweep (CMS) collector

```
-XX:+UseConcMarkSweepGC
```

warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.

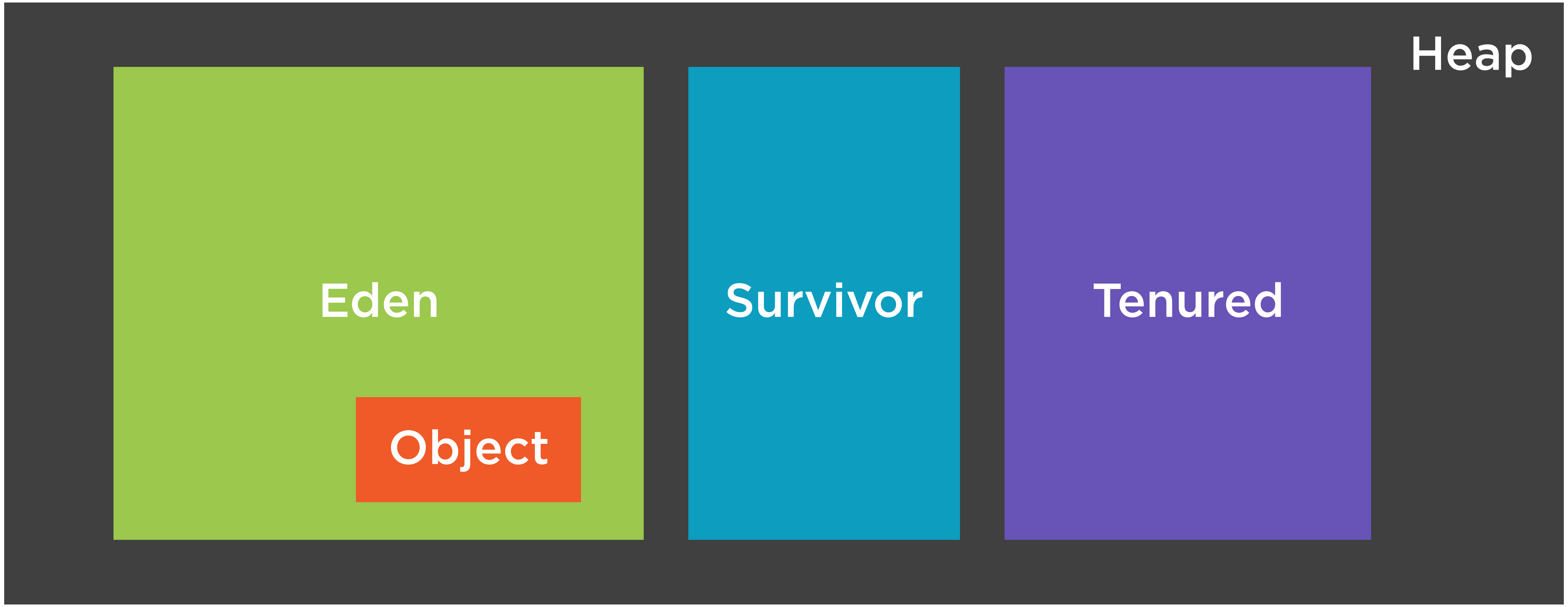
G1 Garbage Collector

G1: Garbage first

Introduced in JDK 6, now default in JDK 9

Replaces Concurrent Mark Sweep GC (CMS)

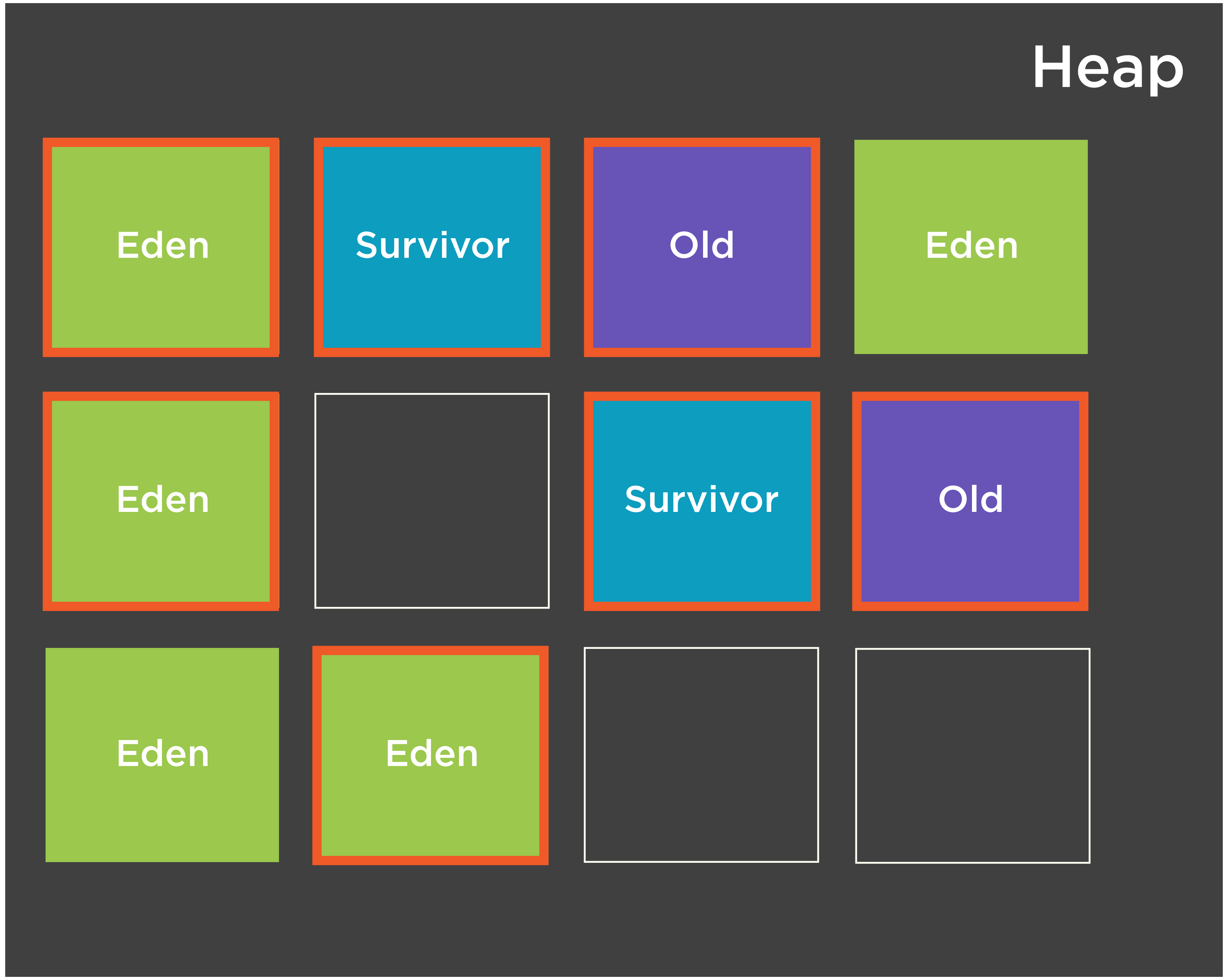
Generational Garbage Collection



Long 'stop-the-world' GC pauses

Difficult to tune

G1 Garbage Collector



Incremental GC

Parallel marking

Designed for large heaps

Low pause, tuneable pause goal

Slightly more CPU intensive

Automatic tuning:

Heap region size

Parallel threads

Pause time interval

G1 Garbage Collector

Faster memory management with fewer code tricks

```
-XX:MaxGCPauseMillis=200
```

Trade some throughput for lower latency

```
-XX:+G1EnableStringDeduplication
```

String Performance

Compact Strings

Lower memory usage

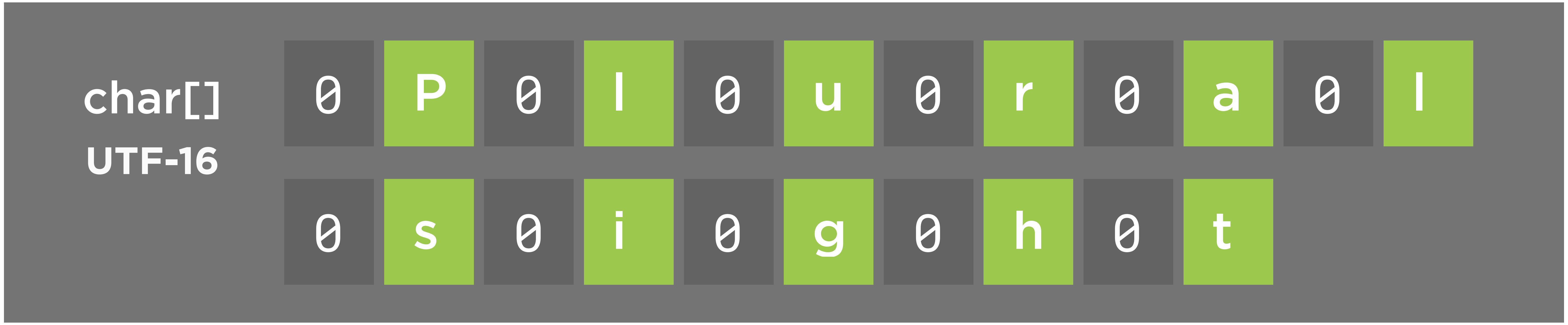
Effective immediately without code changes

String concatenation changes

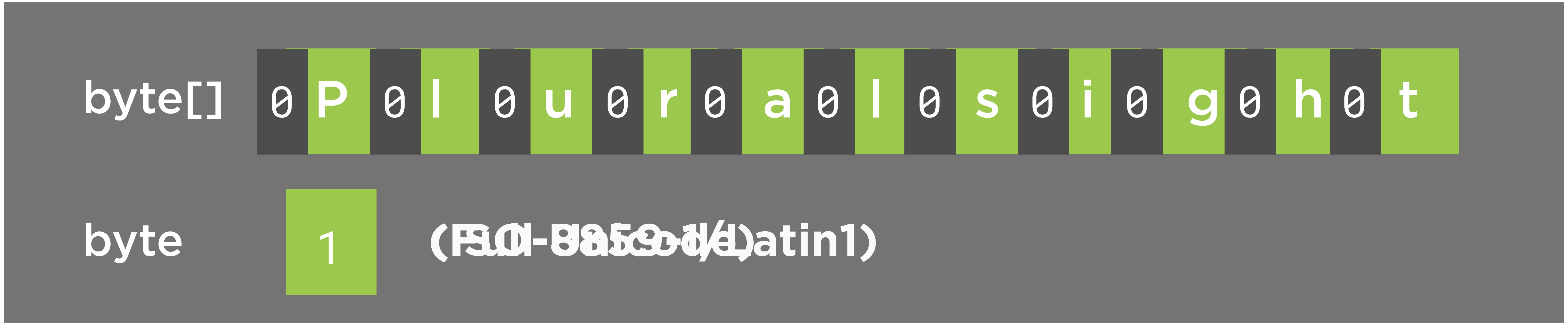
Change concatenation translation strategy

Groundwork for future improvements

Compact Strings



Pre-Java 9



Java 9

String Concatenation

```
String s = "a" + "b" + "c";
```

```
StringBuilder s =
  new StringBuilder();
s.append("a");
s.append("b");
s.append("c");
```

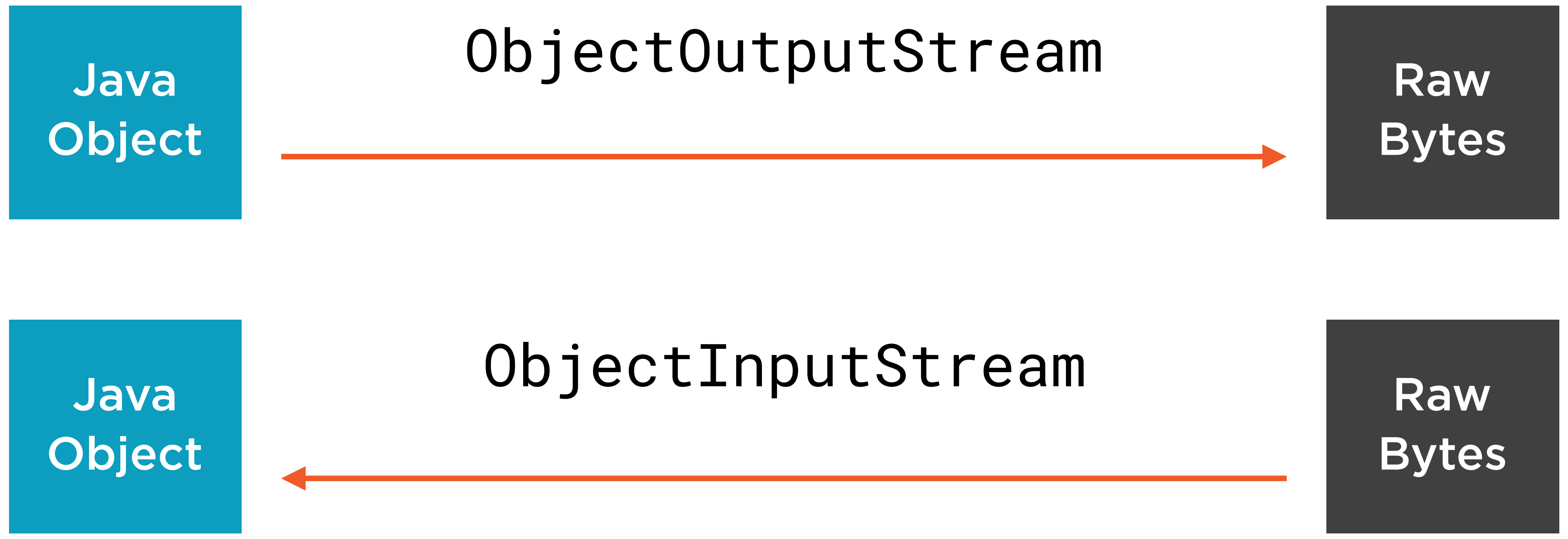
With Java 9:

Invokedynamic bytecode

Late binding to actual implementation

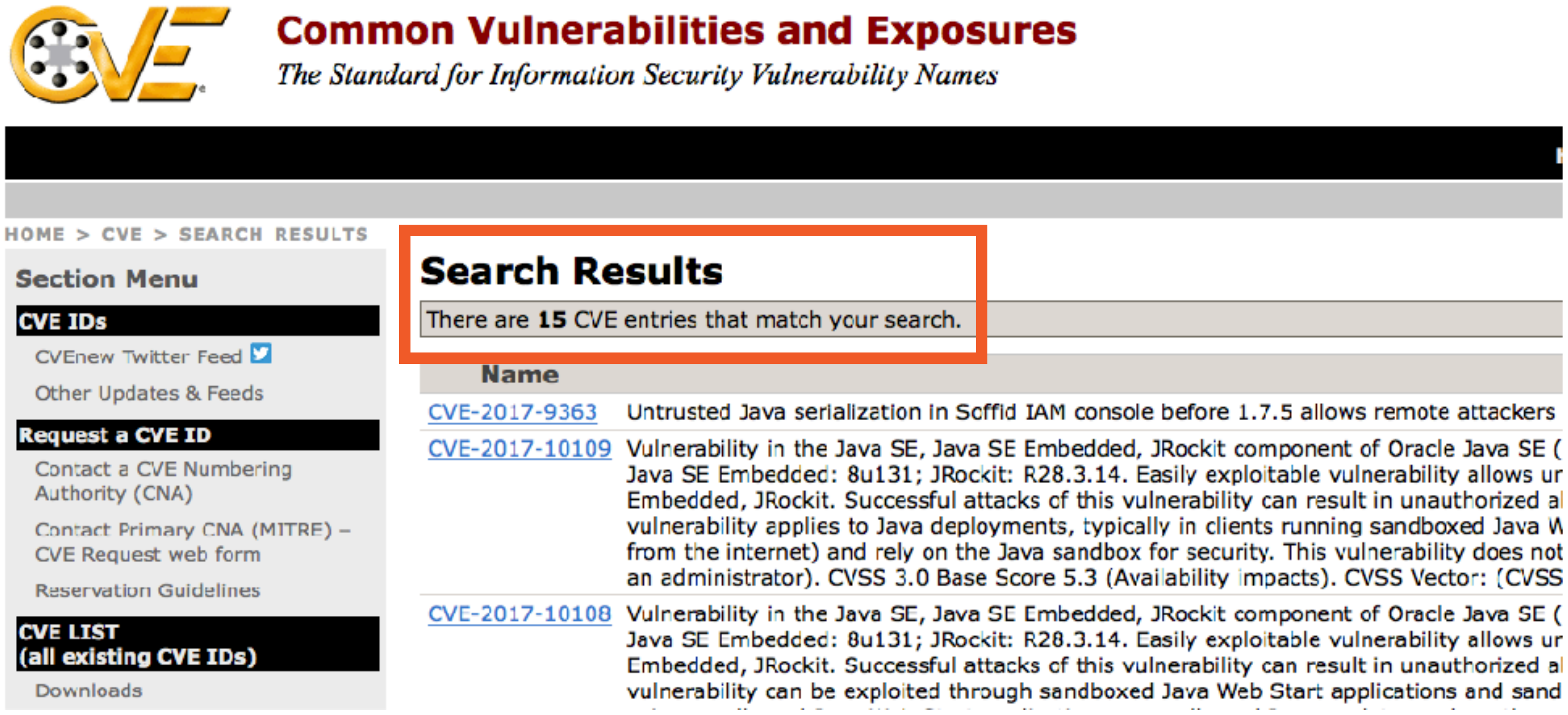
Stable bytecode, future improvements possible

Serialization



Dangers of Serialization

Vulnerability search "Java serialization"



Common Vulnerabilities and Exposures

The Standard for Information Security Vulnerability Names

HOME > CVE > SEARCH RESULTS

Section Menu

CVE IDs

CVEnew Twitter Feed

Other Updates & Feeds

Request a CVE ID

Contact a CVE Numbering Authority (CNA)

Contact Primary CNA (MITRE) – CVE Request web form

Reservation Guidelines

CVE LIST (all existing CVE IDs)

Downloads

Search Results

There are **15** CVE entries that match your search.

Name	
CVE-2017-9363	Untrusted Java serialization in Soffid IAM console before 1.7.5 allows remote attackers
CVE-2017-10109	Vulnerability in the Java SE, Java SE Embedded, JRockit component of Oracle Java SE (Java SE Embedded: 8u131; JRockit: R28.3.14. Easily exploitable vulnerability allows ur Embedded, JRockit. Successful attacks of this vulnerability can result in unauthorized a vulnerability applies to Java deployments, typically in clients running sandboxed Java W from the internet) and rely on the Java sandbox for security. This vulnerability does not an administrator). CVSS 3.0 Base Score 5.3 (Availability impacts). CVSS Vector: (CVSS
CVE-2017-10108	Vulnerability in the Java SE, Java SE Embedded, JRockit component of Oracle Java SE (Java SE Embedded: 8u131; JRockit: R28.3.14. Easily exploitable vulnerability allows ur Embedded, JRockit. Successful attacks of this vulnerability can result in unauthorized a vulnerability can be exploited through sandboxed Java Web Start applications and sand

Filter Incoming Serialization Data

New interface, to filter data before deserializing

```
interface ObjectInputFilter {
    Status checkInput(FilterInput filterInfo);

    enum Status {
        UNDECIDED,
        ALLOWED,
        REJECTED;
    }
}
```

`ObjectInputStream::setObjectInputFilter` per stream

`ObjectInputFilter.Config.setSerialFilter` for all streams

Filter Incoming Serialization Data

Filter without adding or changing code

`jdk.serialFilter` system property:

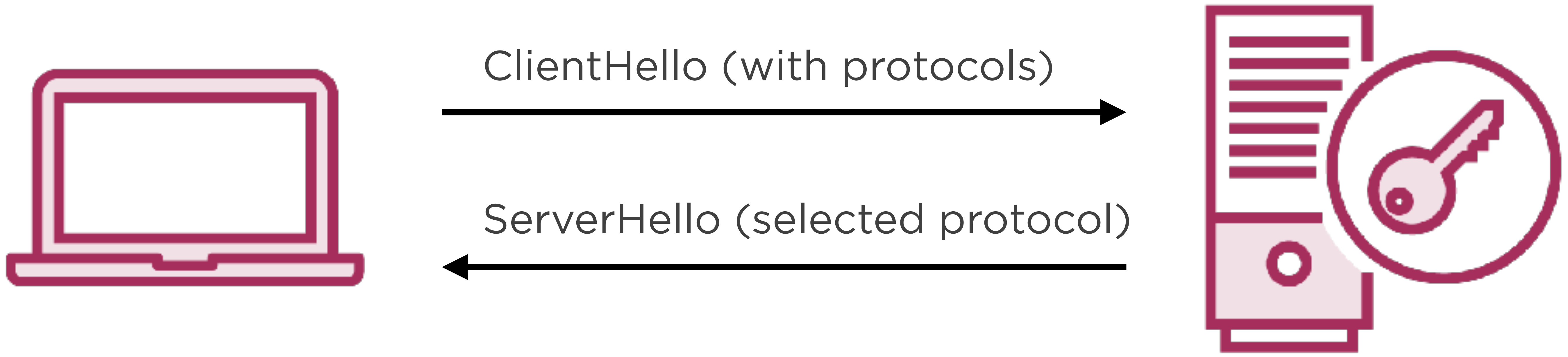
```
maxbytes=n;
maxarray=n;
maxdepth=n;
com.pluralsight.Remote*;
com.pluralsight.dto.**;
!com.pluralsight.internal.*;
```

Backported to Java 6/7/8!

TLS Improvements: ALPN

Application Layer Protocol Negotiation

Select application layer protocol during TLS handshake



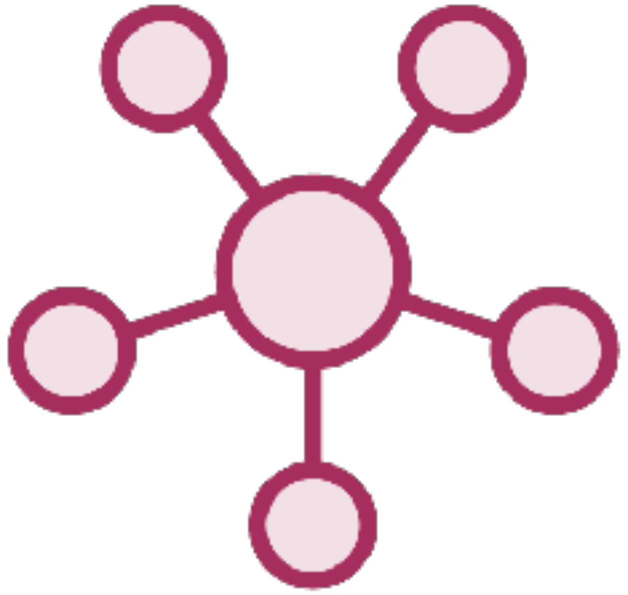
Required for **HTTP/2** support

TLS Improvements: DTLS 1.0/1.2

Datagram Transport Layer Security

Applicable for networking without reliable TCP connection

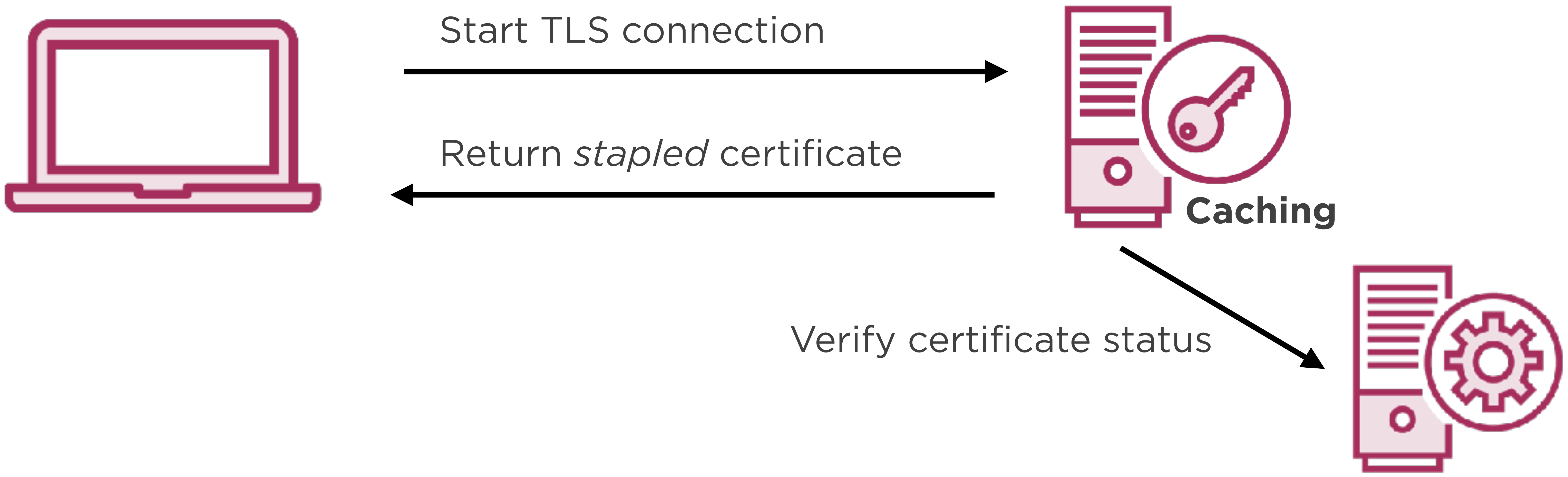
Java 9 supports DTLS 1.0 and 1.2, aligned with TLS 1.1 and 1.2 through `SSLEngine` and `DatagramSocket`



TLS Improvements: OCSP Stapling

Online Certificate Status Protocol

Check for X.509 certificate revocation when establishing TLS connections



SHA-1 Certificates Disabled

SHA-1 'broken' by collision attacks

Certificates using SHA-1 hash rejected by default

Local/enterprise certs not affected

SHA-3 support added

Summary

What’s New in Java 9

Course Wrap-up

Modules

The JDK is modularized, and you can create your own modules as well

JShell

Quick experimentation using snippets of code

Library & Language Improvements

Streams and Optionals got better and more interoperable

Course Wrap-up

New APIs

The ProcessHandle and HttpClient APIs are added

Desktop Java Enhancements

Enhanced rendering performance and a module-ready JavaFX

Performance & Security

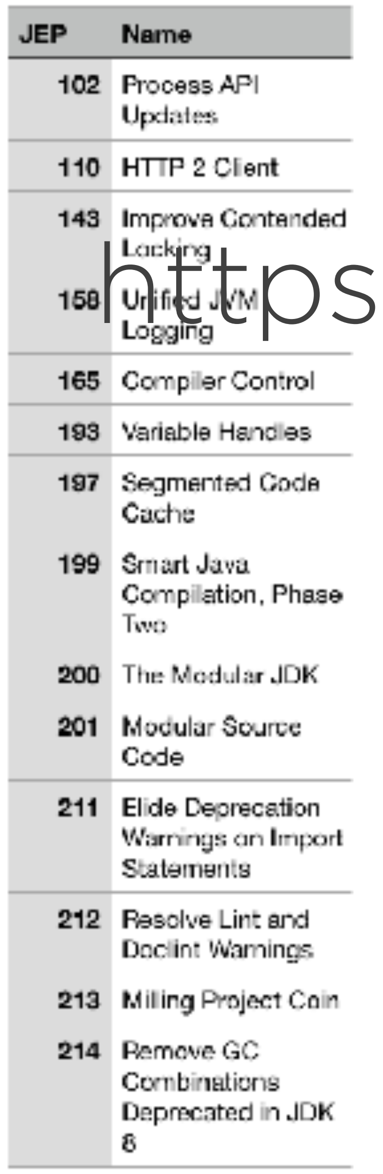
Many TLS related improvements, also to support HTTP/2

Additional Resources

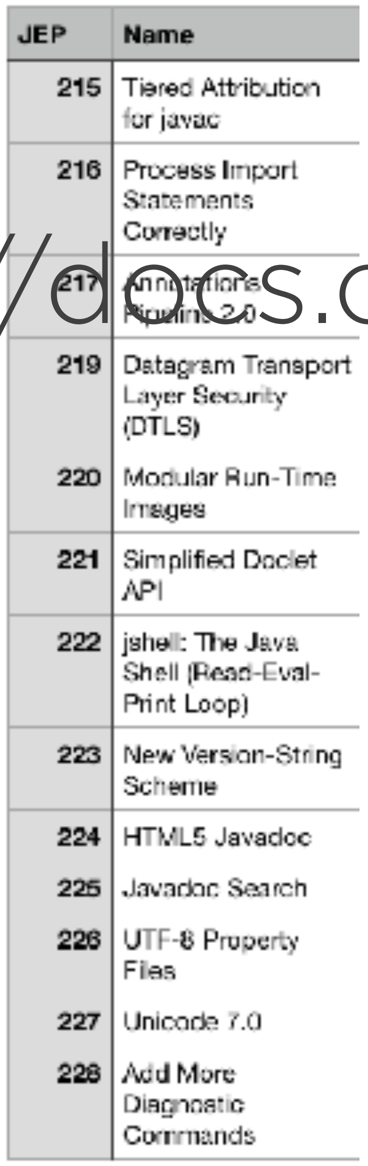
Java Enhancement Proposals (JEPs)

90+ JEPs in JDK 9

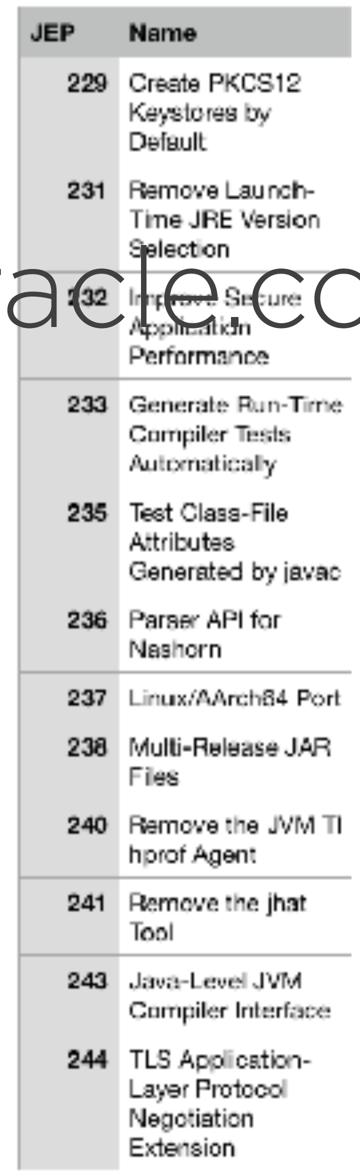
https://docs.oracle.com/javase/9/whatsnew/toc.htm



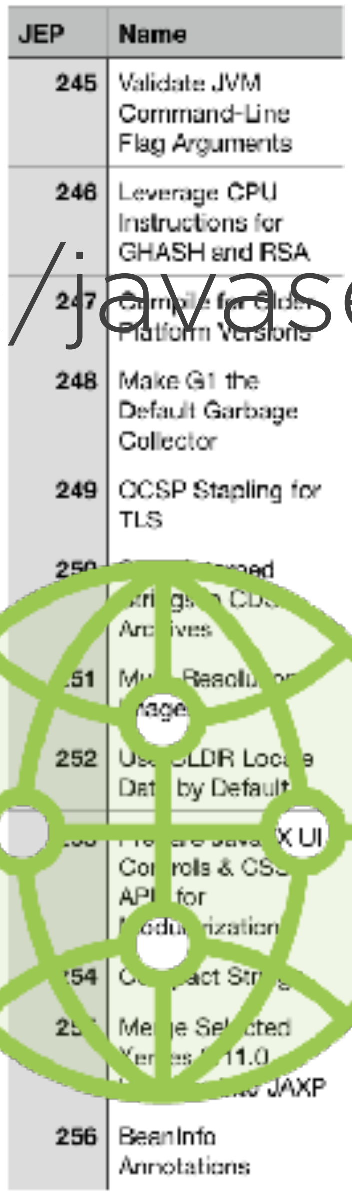
JEP	Name
102	Process API Updates
110	HTTP 2 Client
143	Improve Contended Locking
158	Unified JVM Logging
165	Compiler Control
193	Variable Handles
197	Segmented Code Cache
199	Smart Java Compilation, Phase Two
200	The Modular JDK
201	Modular Source Code
211	Elide Deprecation Warnings on Import Statements
212	Resolve Lint and Doclint Warnings
213	Milling Project Coin
214	Remove GC Combinations Deprecated in JDK 8



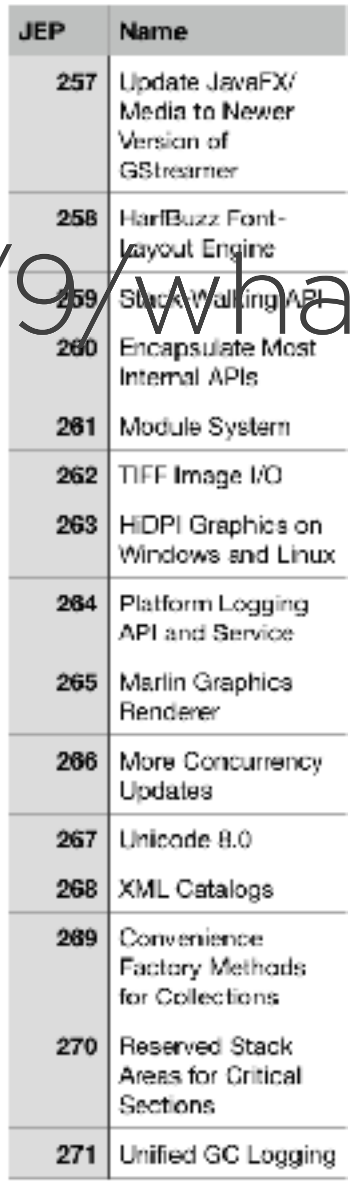
JEP	Name
215	Tiered Attribution for javac
216	Process Import Statements Correctly
217	Annotations Pipeline 2.0
219	Datagram Transport Layer Security (DTLS)
220	Modular Run-Time Images
221	Simplified Doclet API
222	jshell: The Java Shell (Read-Eval-Print Loop)
223	New Version-String Scheme
224	HTML5 Javadoc
225	Javadoc Search
226	UTF-8 Property Files
227	Unicode 7.0
228	Add More Diagnostic Commands



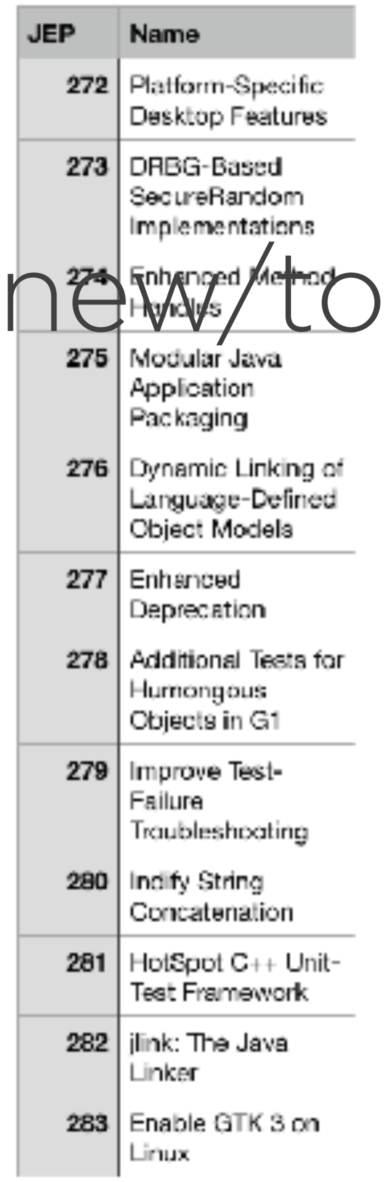
JEP	Name
229	Create PKCS12 Keystores by Default
231	Remove Launch-Time JRE Version Selection
232	Improve Secure Application Performance
233	Generate Run-Time Compiler Tests Automatically
235	Test Class-File Attributes Generated by javac
236	Parser API for Nashorn
237	Linux/AArch64 Port
238	Multi-Release JAR Files
240	Remove the JVM TI hprof Agent
241	Remove the jhat Tool
243	Java-Level JVM Compiler Interface
244	TLS Application-Layer Protocol Negotiation Extension



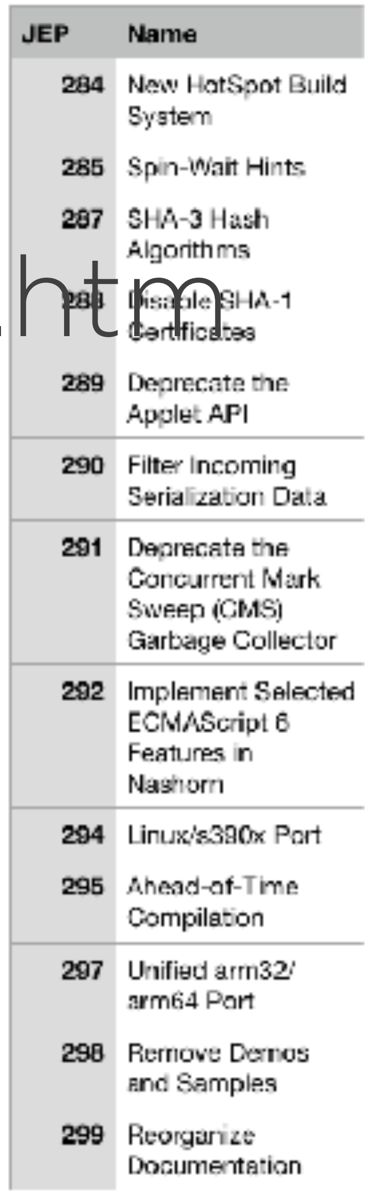
JEP	Name
245	Validate JVM Command-Line Flag Arguments
246	Leverage CPU Instructions for GHASH and RSA
247	Compile for Older Platform Versions
248	Make G1 the Default Garbage Collector
249	OCSP Stapling for TLS
250	Store Interned Strings in CDS Archives
251	Multi-Resolution Images
252	Use CLDR Locale Data by Default
253	Prepare JavaFX UI Controls & CSS APIs for Modularization
254	Compact Strings
255	Merge Selected Xerces 2.11.0 Updates into JAXP
256	BeanInfo Annotations



JEP	Name
257	Update JavaFX/Media to Newer Version of GStreamer
258	HarfBuzz Font-Layout Engine
259	Stack-Walking API
260	Encapsulate Most Internal APIs
261	Module System
262	TIFF Image I/O
263	HiDPI Graphics on Windows and Linux
264	Platform Logging API and Service
265	Marlin Graphics Renderer
266	More Concurrency Updates
267	Unicode 8.0
268	XML Catalogs
269	Convenience Factory Methods for Collections
270	Reserved Stack Areas for Critical Sections
271	Unified GC Logging

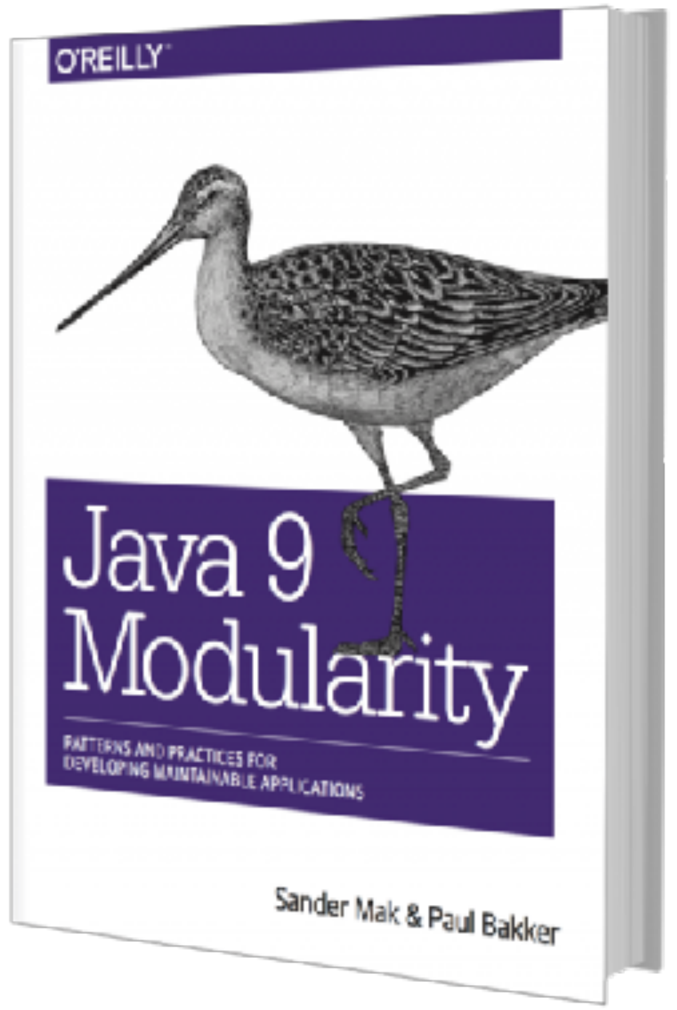


JEP	Name
272	Platform-Specific Desktop Features
273	DRBG-Based SecureRandom Implementations
274	Enhanced Method Handles
275	Modular Java Application Packaging
276	Dynamic Linking of Language-Defined Object Models
277	Enhanced Deprecation
278	Additional Tests for Humongous Objects in G1
279	Improve Test-Failure Troubleshooting
280	Indify String Concatenation
281	HotSpot C++ Unit-Test Framework
282	jlink: The Java Linker
283	Enable GTK 3 on Linux



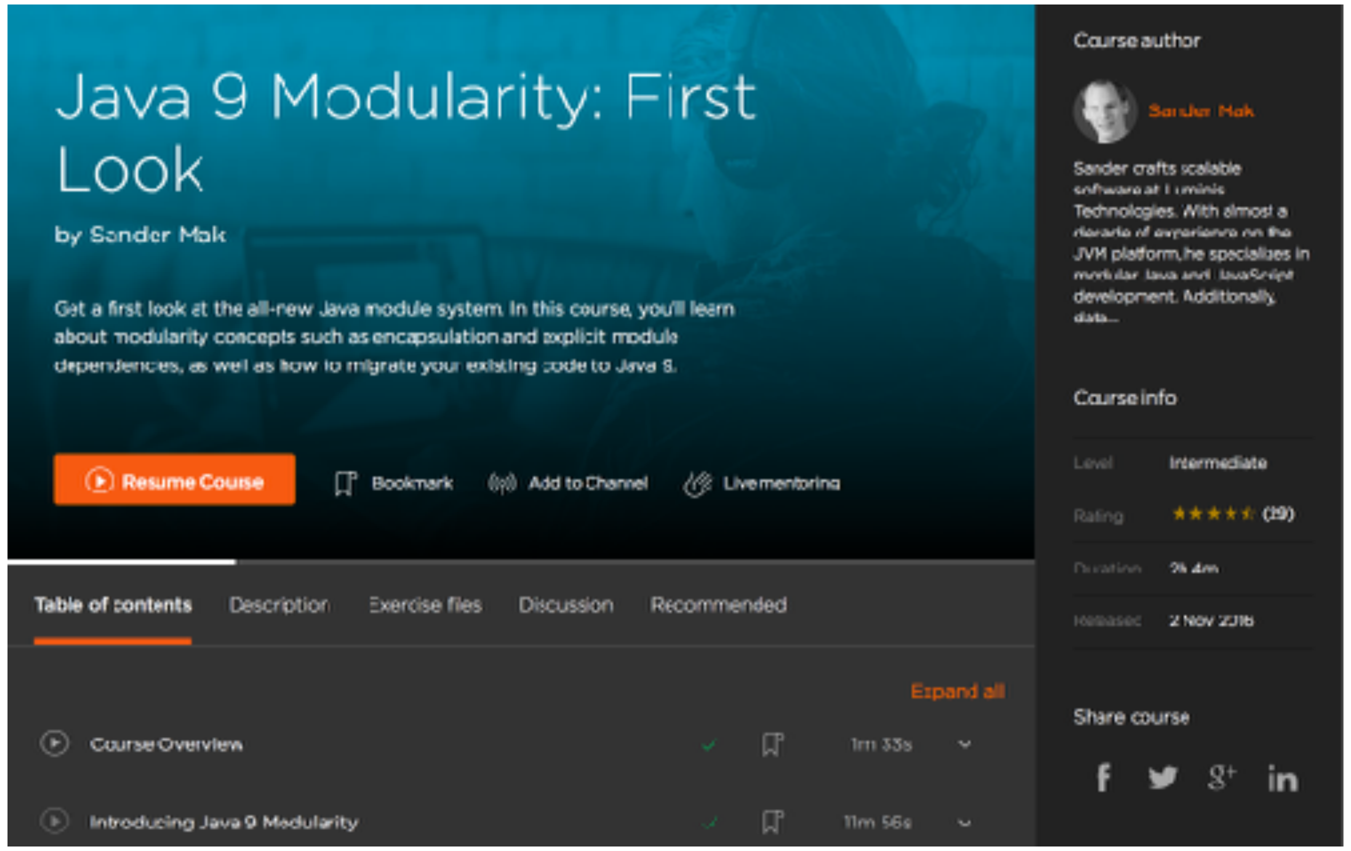
JEP	Name
284	New HotSpot Build System
285	Spin-Wait Hints
287	SHA-3 Hash Algorithms
288	Disable SHA-1 Certificates
289	Deprecate the Applet API
290	Filter Incoming Serialization Data
291	Deprecate the Concurrent Mark Sweep (CMS) Garbage Collector
292	Implement Selected ECMAScript 6 Features in Nashorn
294	Linux/s390x Port
295	Ahead-of-Time Compilation
297	Unified arm32/arm64 Port
298	Remove Demos and Samples
299	Reorganize Documentation

Additional Resources


O'REILLY
Java 9 Modularity
PATTERNS AND PRACTICES FOR DEVELOPING MAINTAINABLE APPLICATIONS
Sander Mak & Paul Bakker

javamodularity.com

Java 9 Modularity: First Look



Java 9 Modularity: First Look

by Sander Mak

Get a first look at the all-new Java module system. In this course, you'll learn about modularity concepts such as encapsulation and explicit module dependencies, as well as how to migrate your existing code to Java 9.

Course Overview 1m 33s

Introducing Java 9 Modularity 11m 56s

Course author

Sander Mak

Sander crafts scalable software at Luminis Technologies. With almost a decade of experience on the JVM platform, he specializes in modular Java and JavaScript development. Additionally, data...

Course info

Level	Intermediate
Rating	★★★★☆ (29)
Duration	2h 4m
Released	2 Nov 2016

bit.ly/java9course

@sander_mak